The Combination of Investment Strategies Using the Replicator Equation.

Author:
Saskia van der Wegen

Company supervisor:
ing. Joop Snijder

First supervisor:
dr. Gerard Vreeswijk

Second supervisor:
prof. dr. Frank Dignum

Master's thesis
Department of Information
and Computing Sciences
Utrecht University
October 4, 2019

Utrecht University

This research has been conducted in collaboration with Info Support and PGB Pensioendiensten. I would like to thank my company supervisors Joop Snijder (Info Support) and Hemir Brand (PGB) for their great support.

Abstract

Recently, there has been a lot of interest in applying artificial intelligence to predicting future stock prices. One of the studied approaches makes use of multi-agent models to make those predictions. The current study is built upon this previous research that uses multi-agent prediction models. It answers the question whether a dynamic multi-agent model using the replicator equation to evolve a population of trading strategies over time can be used to combine stock index trend predictions. Of course, as a first step such a model is build. This model consists of several different trading strategies that all make up a certain part of the population. The replicator equation is used to evolve the proportions of those individual strategies in the population. Thereafter, the models ability to predict future trends of the S&P500 is tested. The model is not shown to be able to make good predictions. It is concluded that the replicator dynamic is too slow for this purpose. Thereafter, a second experiment is conducted with a revised prediction model. In this experiment largely the same model and method is used as in the first one. However, the revised model makes use of the ReDVaLeR algorithm to evolve the system, instead of the replicator equation. No good prediction performance is found from this revised model as well. Therefore, no evidence is found that it is useful to combine multiple investment strategies using a dynamic multi-agent model based on the (modified) replicator equation.

Contents

1. Introduction

The Pension Fund for Graphic Companies (PGB) is a Dutch pension fund for employees in multiple sectors, such as graphic media and sea fishery. As all pension funds, they aim to invest their money in a way such that profits are maximized. They do this by buying and selling stocks and bonds. Therefore, they face the difficult problem of optimal portfolio management.

To achieve good results on the stock market, it is important to be able to predict future stock prices. After all, when a stock will become worth more in the future, you can benefit from this by buying that stock. On the other hand, when the value of a stock you own will decrease in the future, now is a good moment to sell.

The problem is, that predicting future stock prices is not that easy. In fact, it is extremely difficult! The difficulty is that stock prices are entirely based on supply and demand. When the demand for a certain stock is higher than its supply, the price will raise. If the opposite is true, the price will fall. Consequently, the problem of predicting stock prices, is a problem of predicting supply and demand.

Supply and demand are, of course, depending on consumer behaviour. In this case, the trading behaviour of investors. Therefore, predicting this behaviour could help predicting future stock prices. This has let to PGB being interested in building a model to simulate the behaviour of other investors, in order to help predicting future stock prices.

This interest does not come out of nothing. For a long time, the efficient market hypothesis has been the most prominent paradigm in theoretical finance (Sorropago et al., 2014). This hypothesis states that asset prices fully reflect all available information (Malkiel & Fama, 1970). It does follow from this hypothesis that asset prices are unpredictable. However, more and more people come to question one of the assumptions underlying this efficient market hypothesis (Sorropago et al., 2014). Namely, the assumption that traders act rational . Instead, the field of behavioural finance is growing. This field studies the, not always rational, behaviour of real traders. Lots of knowledge about the behaviour of real traders is gathered by now. Consequently, people become to wonder if, and how, this information can be used to help asset price prediction.

There are already models that try to simulate the behaviour of investors in order to predict stock prices (e.g. Gou, 2005; Gupta, Johnson, & Hauser, 2005). In some of those studies, the stock market is simulated as a multi-agent game.

The agents in those games represent traders with different (sets of) trading strategies. Those agents can choose to sell or buy stocks (and sometimes also to do nothing). Based on those actions, future stock prices are predicted.

There are two important decisions that have to be made when developing such a multi-agent system (Gupta, Hauser, & Johnson, 2007). The first important decision concerns the architecture of the multi-agent model. This includes the specification of both the agents and the market mechanism. The second important decision when building such a multi-agent market model concerns the specification of the proportions of the different types of agents in the population. To limit the scope of this study, we will only focus on this second decision.

At the real financial market different traders use different trading strategies (Farmer & Joshi, 2002). Those different traders can be simulated as different groups of artificial agents, where every group has its own strategy. Those artificial agents could be designed to describe real traders as good as possible, but could also differ from real traders. This depends on the goal of the simulation. In such multi-agent simulations, it is not known beforehand how many of the market participants use each strategy. Therefore, the proportion of each group of agents should be learned.

Some research that aimed to find the proportions of each agent type to best predict future stock prices is already done (e.g. Gupta, Hauser, & Johnson, 2005). This research made use of either minority, or majority games. Depending on the game, the goal of the agents is to choose the action respectively the minority, or the majority, of the agents chooses. In those studies, there are multiple groups of agents. Those groups are characterised by the different decision making strategies they use. The proportions of those groups are set to optimally predict the real data.

Although some of those models have some early success in predicting stock prices, the models don't represent real agents very well. Not in the last place, because they generally assume the composition of the trader population to be static. That is, they assume that the proportions of the different types of agents stay stable over time. However, this assumption can be questioned. People who lost money when maintaining their current strategy could choose to temporary retract from the market or to adapt their strategy. Furthermore, those people recently lost some money, which may result in a limited capital for future investments. As a consequence, those people will have a smaller effect on future price changes. Traders who recently earned money, on the other hand, have more money to invest. Furthermore, others can decide to use their winning strategy as well. Therefore, it is more realistic to assume that the proportion of each strategy in the population changes over time.

It is suggested that representing the structure of the real system more realistically in the agent-based model, can enhance the performance of the model in explaining and predicting real-world phenomena (Edmonds & Moss, 2001). Therefore, dynamically updating the proportion of each strategy in a multi-agent model could possibly enhance its prediction performance. At this moment, such dynamic forecasting models do not yet exist.

There are some descriptive studies that change the proportions of different types of agents over time, but those use artificial market models to simulate the market (e.g. Katahira, Chen, Hashimoto, & Okuda, 2019). In those artificial market models, prices are set endogenously in the model, so no real price histories are used. Therefore, those models are suitable to reproduce and study some qualitative market features, but are not suitable to forecast real prices.

Forecasting real prices is, to the best of our knowledge, not yet done with such a dynamic multi-agent model. Therefore, in the current research it is studied whether a multi-agent model in which the proportions of each type of agent are dynamically updated over time, can be used to combine investment strategies that predict the trend of stock index prices.

As this is an exploratory study, we do not aim to compete with state of the art trend prediction models. Rather we aim to find out whether using a dynamic multi-agent model is a useful method to combine investment strategies. Instead of making a perfect dynamic multi-agent model at once, we aim to proof the value of such a model with a simplified one. The model is considered useful when (1) it performs better than all the strategies it combines, (2) it performs better than an easy model with static proportions, and (3) its performance is robust. That is, its performance is stable over time and does not highly depend on accidental initial values. This is explained in more detail later on.

As our model is constantly updated over time, it is expected that it can handle the dynamic characteristic of the financial market better than static models. The financial market changes constantly, which means that a strategy that performs very good at the current moment, is not guaranteed to perform as good at later moments in time (Timmermann, 2006). Due to its dynamic characteristic, our model is expected to be able to handle such a changing market and outperform static strategies.

To see whether this expectation is true, we implemented a multi-agent system with trading agents. Each agent in the system uses its own trading strategy. The proportions of the agents in the population are based on the success of those agents in predicting the price movement. After all, successful agents have more money to invest in the future and therefore a larger impact on future price changes. Furthermore, agents that use a losing strategy can choose to switch to a winning strategy. Therefore, our model assumes that agents using successful strategies will make up a larger part of the population than those using unsuccessful strategies.

After building the model, it is tested whether the model is able to make correct trend predictions. To do this the performance of our model is compared to the performance of the strategies it combines and a model with stable proportions. Furthermore, the models ability to handle changing environments and its robustness are studied.

2. Literature

Before we explain our dynamic multi-agent forecasting model, it is important to have some background knowledge. In this section a small, though by far not exhausting, overview of existing stock market prediction methods is given. Thereafter, we give some insight into the state-of-the-art multi-agent market models. Last, it is described how we can integrate those into a new, dynamic multi-agent forecasting model.

2.1 Stock market prediction

There are, of course, already lots of models that aim to predict stock prices. Those can be divided into two categories. On the one hand, there are models that make quantitative predictions of future stock prices (Montgomery, Jennings, & Kulahci, 2015). Those are generally using some form of regression analysis. Some important examples are vector autoregressive models (VAR), autoregressive integrated moving average models (ARIMA) and various types of neural networks (NN). However, other types of quantitative models are possible as well. Examples of this are smoothing models and general time series models.

On the other hand, there are classification models. Those do not aim to predict the exact future price, but only classify the expected direction of the time series as upwards or downwards. Of course, models that make quantitative predictions can be used to predict the trend as well. However, there are also models that explicitly aim to predict the trend and make no quantitative predictions at all. Important examples are k-means clustering and support vector machines (Maharaj, D'Urso, & Caiado, 2019).

Besides qualitative and quantitative models, another distinction between forecasting methods can be made. This distinction is not based on the type of output of the method, but on the type of information the forecast relies on. The two main types that can be distinguished are technical analysis and fundamental analysis (Beyaz, Tekiner, Zeng, & Keane, 2018).

Technical analysis uses historical data of stock prices and trading volumes to predict future prices. So this type of analysis solely looks at past market activity and does not care about the company underlying the asset. This type of analysis is based on the idea that previous patterns of price movement are likely to repeat themselves. Therefore, technical traders are looking for patterns in historical price data, and try to extrapolate those patterns to future prices.

Fundamental analysis, on the other hand, looks at the company underlying the asset itself and tries to estimate its intrinsic value. This is done by looking at signs from both the overall economy and the specific company of the asset. Examples of such sign are the profitability of the company, and the overall state of the economy.

There is one important assumption underlying this fundamental analysis. Namely, the assumption that prices in the long run converge to the fundamental price of the company. So when an asset is undervalued at the moment, a trader is expected to earn money buying that asset, as the price will rise to the intrinsic value of the company.

Both fundamental and technical analysis have proven to be able to make correct predictions in some situations. However, there are many different fundamental and technical indicators. It goes too deep to describe all of those here, but it may be clear that the big question is which one is the best to use in order to earn as much money as possible. Unfortunately, this question does not have a straightforward answer. Which one is best to use depends on many factors, like the stock that is invested in and the time (Lee, 2011).

As no single indicator can be pointed at as the best one, often multiple different indicators are combined in order to get the best of all. Therefore it is not surprising that there exist many different forecasting models combining multiple indicators or even multiple forecasting strategies (that can itself use multiple indicators) (Timmermann, 2006). They can not only combine various technical indicators or various fundamental indicators, but can also make a combination of indicators from both types of strategies. Those combined models are further described in the section below.

Combination models

As said, there are many different ways to combine multiple forecasts into one, hopefully more accurate, forecast. As Lee (2011) states, it is difficult to find one forecasting method that outperforms all other methods over a long time range. Therefore combined models are used to improve the forecast accuracy, and not without success (Huang & Lee, 2010).

Such a combination has several advantages over single forecasting methods (Timmermann, 2006). First, the performance of such a combined model is more stable than that of a single strategy. At this moment, no single strategy can capture the market perfectly well, and therefore, no single strategy can predict it perfectly well. However, the point where those strategies specify the marked wrongly is different for all those strategies, therefore combining them can help making better predictions, as when one strategy goes wrong in a certain situation, others can still do well in that same situation. So the performance over time becomes more stable, and therefore less risky, than the performance of a single strategy that differs much more over time.

Another advantage of combining models is that the model choice itself is less risky. When choosing one single strategy, the consequences of choosing a wrong one can be tremendous. When combining models, this risk is reduced, as there are other strategies in the combination that can compensate for the bad strategy.

Because of those advantages, it is not surprising that there is done lots of research into optimal ways to combine strategies. This combining is most often done by taking the weighted average of all different strategies. But how do you decide those weights?

A very successful method is to use just equal weights. This may seem a bit too easy, but has been proven to be very effective and hard to beat with more sophisticated methods (Timmermann, 2006). Equal weights work especially well when all strategies are, by approximation, equally good.

Lots of other possibilities are studied as well. One such a method, which knows lots of variations, is least-squares regression. In this method the weights are chosen such that the squared errors between the predictions and the real prices are minimized.

Another successful method is trimming (Timmermann, 2006). With trimming the few worst performing strategies are eliminated. Thereafter, the weights can be calculated with a method of choice, for example equal weights or least-squares. This has been shown to significantly improve the predictions.

Another method that can improve predictions is using time-variant weights. This leads to combination models in which the weights of each strategy in the combination vary over time. Its usefulness is very intuitive; as the performance of each strategy changes over time, so should its weight. And indeed, such time-variant models can sometimes outperform stable models (Elliott & Timmermann, 2005), although this is not always the case.

There are various different ways to update weights dynamically over time. One popular method makes use of rolling time windows (Timmermann, 2006). In this method a time window of length n is used, where n is a fixed number of time steps. The optimal weights are then estimated over the n last time steps. So every time step, new optimal weights should be estimated. This is either done by regression or another weight estimation method.

Another method to adjust weights over time is to use expanding windows, instead of rolling windows. Herein, the length of the history on which the weights are estimated is expanded. This can, for example, be done by updating or recomputing the weights using regression. Of course, there are many other possibilities as well. However, as the literature on the topic of forecast combination is far too fast to give a complete overview here, the reader is referred to Timmermann (2006) for a more detailed overview of the different combination methods.

2.2 Agent-based models

Above, a lot of prediction methods are described. A common characteristic of all those types of methods is that they directly try to fit the financial time series. They do not look at the underlying system, that causes the prices to change. That is, they are not interested in the interactions between supply and demand. This makes it hard to interpret and explain the outcomes of the model (Grothmann, 2003). Using models that predict future prices by simulating the behaviour of real investors can lead to results that are more easy to analyse and interpret (LeBaron et al., 2001). Multi-agent prediction models can be used for this task.

The literature related to stock price prediction using multi-agent models can be divided into two parts. On the one hand there are the existing models that use agent-based simulations to predict stock prices. Specifically those that learn the composition of the population. On the other hand there are the existing studies that model the proportions of different types of agents dynamically. Below, those two lines of research are discussed.

Predictive models

There are various studies that use multi-agent systems to predict stock prices. Some of those are explicitly focussed on finding the optimal probability distribution over the different groups of agents in the model (e.g. Gupta, Hauser, & Johnson, 2005; Thimmaraya & Masuna, 2012). In this context, the optimal probability distribution is defined as the distribution with which the predictions of the model fit the real data as good as possible.

One way in which this is done is by translating the problem of stock price prediction into a multi-agent game. An often-used game is the minority game (Gupta, Hauser, & Johnson, 2005). In this game, agents choose to either buy or sell assets. If the majority of the agents chooses to buy assets, the price will rise. On the other hand, if the majority chooses to sell, the price will fall. In that way, the direction of changes in asset prices can be predicted.

One of the first researchers introducing the minority game were Challet and Zhang (1998). Later, the game is applied to financial time series prediction by multiple researchers, like Gupta, Hauser, and Johnson (2005). In the game, multiple agents with limited resources compete with each other. They have two options to choose from: buying and selling a number of assets. In some advanced versions, there is the third option of doing nothing. The goal of the agents is always to make the minority decision. That is, taking the action that the minority of the agents takes.

Later, this game is extended to a mixed minority-majority game (Marsili, 2001). Marsili (2001) argues that traders are heterogeneous. The two main groups of investors he distinguishes are fundamentalists and trend followers. The fundamentalists expect prices to converge to some fundamental value. So if an asset is overvalued, they expect the price to fall, and the other way around. Trend followers, on the other hand, expect current trends in asset prices to persist. So if the price is rising, they expect the price to stay rising in the future.

As Marsili (2001) argues, minority games only represent fundamental traders. The trend followers, however, play an entirely different game. When the majority of agents sells an asset, they expect the price to fall, so selling is the optimal strategy. Therefore, it can be argued that trend-followers play a majority game. This led to the development of mixed majority-minority games.

Those games are, amongst others, used by Gou (2005) to predict financial time series. The game is similar to the minority game, with the difference that some agents play a majority game instead. What the models using the minority game and the mixed game have in common, is that the proportion of the different types of agents in the population can be inferred from real market data.

In, for example, the study of Gupta, Hauser, and Johnson (2005), each agent has a set of strategies to choose from. Through experience, they learn which of those strategies gives the best performance. However, the number of agents using each set of strategies is not fixed beforehand. This is fitted to the data in order to get a good match with real market data. They do this by means of a Kalman Filter. This is a least-squares optimization method that tries to iteratively find the best estimate of the proportions of each strategy set. Although the proportions are learned iteratively, in the existing studies they remain fixed after the initial learning period.

A totally different study makes use of particle swarm optimization to find the optimal probability distribution over agents or strategies (Thimmaraya & Masuna, 2012). In this study, three different types of agents are used. Those are long term investors, short term speculators, and small, random investors. Those agents all make investment decisions. Those decisions are combined to make a prediction of the future price of gold. The weight each of those agents has in this combination is decided using particle swarm optimization. The weights are chosen such that the prediction error of the model, compared to the real price of gold, is minimized.

Adaptive models

The studies described above calculate the proportions of each group of agents only once and thereafter use those proportions to make predictions. However, it would be more realistic to assume that those proportions change over time (Gupta, Johnson, & Hauser, 2005). To the best of my knowledge, there are no studies yet that use such an adaptive multi-agent model to predict real time series. There are some studies in which the proportions of each group of agents is adaptively changed. However, instead of using real data, those studies make use of an artificial market model (e.g. Galla & Zhang, 2009).

Galla and Zhang (2009) use a multi-agent game in which each agent has only one strategy on which it decides its action. The difference with earlier mentioned models, is that the capital of the agents changes dynamically. So if the agent was successful in the past, its capital has grown, while the capital of unsuccessful agents decreased. The larger the capital of an agent is, the more it can invest. Therefore, rich and successful agents have a larger impact on future price changes than poor agents.

Another difference with the aforementioned studies, is that success in the study of Galla and Zhang (2009) is not defined by external price data (the data of real prices), but by an internal market mechanism. That is, the price is part of the model as well, and will be updated in accordance with the actions of the artificial agents. So if agents have a larger capital, their decision will have a larger impact on the change in price than the decision of agents with a small capital. This model is shown to be able to reproduce some qualitative characteristics of the financial market.

2.3 Discussion

In the first section of this chapter, we have seen that combining forecasts can improve the overall forecast quality. Especially calculating a weighted average with time-varying weights is shown to work well. On the other hand, we saw that multi-agent models can be used to predict future stock prices. This can also be seen as a combination model, after all predictions of multiple agents are combined. However, weights or proportions of agents are not yet updated over time in existing multi-agent models. Indeed, there is at least one multi-agent model that does update the weights over time, but this uses an artificial market model instead of real price histories.

All in all, there are no multi-agent models that aim to predict stock prices, with time-varying proportions yet. Therefore, in this study we research the possibilities of such a model. Such a model is expected to combine the good things of both time-variant combination forecasting models and multi-agent prediction models. Those are respectively high performance and high explainablility.

To achieve this, we build a multi-agent model, but dynamically change the proportions of each type of agent over time. Of course, we can do this using an optimization method as described in Timmermann (2006). However, those methods do not fit the multi-agent structure of our model. In those optimization models, the combination of strategies is directly optimized. That is, the optimal composition of the population as a whole is calculated. However, it is not realistic to assume that all agents in the population work together to find the optimal division over strategies. It is much more realistic to assume that all agents try to maximize their own returns (by choosing the best strategy) and do not care about the others. Therefore, the end product or prediction will simply be a combination of all those individual acts. Holding this multi-agent aspect in mind, we update the proportion of each agent using some population dynamic and see whether this can be used to make good predictions as well.

How this is done exactly is explained in chapter 4. First, in chapter 3, the used multi-agent model is explained, to get a better understanding of what is going on in the model. Thereafter, the performance of the model is evaluated and some possible improvements are discussed.

3. Multi-agent model

To study whether a dynamically updating multi-agent model can have an added value for predicting future stock prices, we first need to define such a model. This model is described below.

3.1 Model definition

In this study, a multi-agent model is developed in order to predict financial time series. At every step in time, the agents in the model observe the historical price data of a stock index. This data contains the daily closing prices of the index. Furthermore, the agents observe the return rate of a risk-free investment. Based on those data, the agents can choose between three actions: buying the index fund, selling the index fund, or investing in a risk-free obligation. This last possible action is similar to putting the money away at a risk-free rate of interest. After performing the action, the closing price of the index on the next day is observed. Based on this price the rate of return of each agent is calculated. Hereafter, the whole process of perceiving the price history, choosing actions and observing the next price is repeated. This whole process can be seen in Figure 3.1.

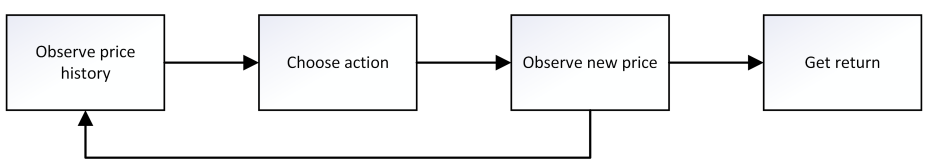

Figure 3.1: The process for an individual agent.

The artificial agents in the model differ in the trading strategy they use. The agents are divided into different types, where each type of agent uses one specific trading strategy. The proportion of each of those agent types in the population of all agents is used to make predictions of future stock prices.

At every step in time, the agents choose between the three possible actions. Based on the combination of the actions of each agent, the future direction of the stock index price is predicted. The decisions each type of agent makes are added, while taking into account the proportion of the population that consist of this type. When the index fund is in our population of artificial agents bought more often than sold, the real price is predicted to rise. When the index fund is sold more often, the price is expected to fall. This is summarized in Equation 3.1 (based on Antonello & Silveira, 2010).

$$\hat{p}^{t+1} = sgn(\sum_{i=1}^{N} x_i^t a_i^t), \tag{3.1}$$

where $\hat{p}^{t+1}$ is the predicted direction of the change in price between time t and t+1. This is 1 if the price goes up, 0 if the price stays equal and -1 if the price is predicted to fall. N is the number of different types of agents, x_i^t is the proportion of agent type i at time t, and a_i^t is the action agents of type i chose at time t. The action selling is represented by -1, risk-free investing by 0 and buying by 1. This entire process of stock price prediction for one time step is summarized in Figure 3.2.

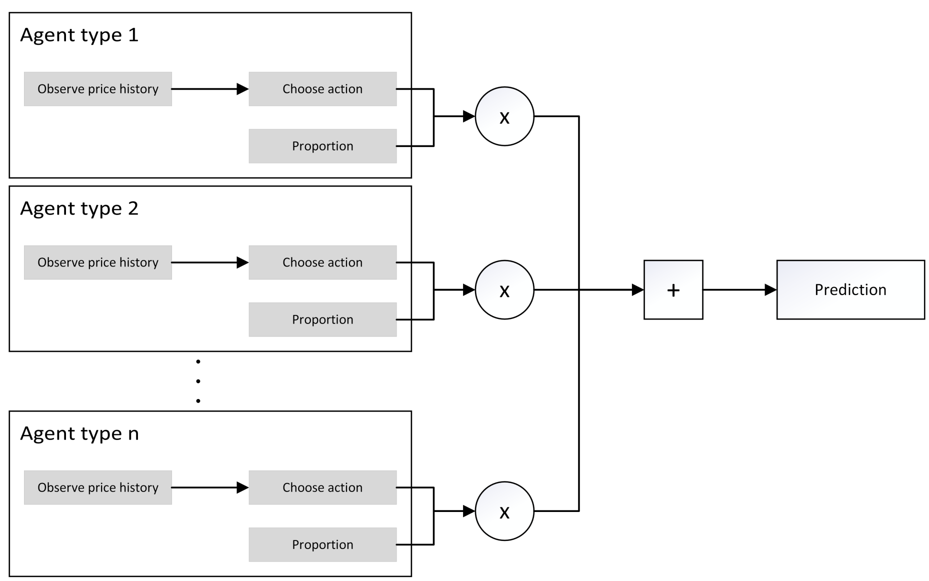

Figure 3.2: The combination of individual actions into an overall prediction.

After predicting the stock index price on the next time step, the real price on that time step is observed. This observation does not only give certainty about the success of the total prediction, but can also be seen as a measure of the quality of the individual agents' prediction. Agents that predicted the price change correctly gain money while others lose money.

Therefore, those agents that were succesfull have more money to invest, and as a consequence, from now on form a larger part of the population. Furthermore, (artificial) traders that loose money, can choose to switch to another trading strategy to have more success in the future. Therefore, the proportion of each type of agent is updated over time, based on its success. How this is done is discussed in the next section.

3.2 Weight estimation

Now we have defined the workings of the multi-agent model, the remaining question is how the proportions of the different types of agents are estimated. There are many possible ways to do this. As describe in chapter 2, there are already lots of studies into integrating different forecasting strategies. However, those do not fit our multi-agent approach, as they do not describe how individual agents grow their capitals and choose between strategies, but instead only try to optimize over the population of strategies as a whole. In our study, we want to see if good forecasts can also be made using more realistic population dynamics. There are multiple ways to evolve the proportions of agents in the population over time. Two of those are describe in the subsequent sections.

3.2.1 Capitals

One of the most obvious methods to decide the proportion of agents in the total population is to set the proportion of a type of agent equal to its capital divided by the total capital in the population. This method is already used in artificial market models to evolve the composition of agent populations (e.g. Galla & Zhang, 2009).

Agents with a large capital have more money to invest and therefore have a larger effect on the future price changes. Assuming that agents always invest the same proportion of their capital, regardless the size of that capital, the influence of the agent on the future price will grow linearly with its capital.

As capitals do not stay static, but change in accordance with the trading success of the agent, the proportions of the different agents change over time as well. The capital of an agent that gains money by predicting the stock market correctly will grow, while the capital of a losing agent will fall. More concrete the capital evolves using Equation 3.2 (based on Galla & Zhang, 2009),

$$C_i^{t+1} = max(C_i^t * (1 + \epsilon * r_i^t), 0), \tag{3.2}$$

where C_i^t is the capital of agent i on time t, ϵ is the proportion of the capital that is invested and r_i^t is the percent of its invested money that agent i earned (or lost) at time t. This percentage corresponds to, depending on which action is chosen, the return rate of the predicted time series at time t or the risk-free return rate on this time. Assuming capitals cannot become negative, the maximum of the new capital and zero is chosen.

The proportions are thereafter calculated as follows:

$$x_i^t = \frac{C_i^t}{\sum_{j=1}^{n} C_j^t}, \tag{3.3}$$

where x_i^t is the proportion of agent i in the population at time t, and n is the number of agents.

While this method of evolving capitals fits artificial market models very well, it can be questioned whether it fits that well when used in combination with external price data. In artificial market models, the change in price depends on the actions of the individual agents. So agents with a larger capital can invest more and therefore have a larger influence on future prices. However, in our model, prices are set externally (by the real market) and do not directly depend on the actions of our artificial agents. Therefore, growing capitals are not as self-evident as a good population dynamic when using external data as when using endogenously determined price data.

Indeed, the method is shown to have some disadvantages when used on real price data. The most important one is that capitals grow exponentially using this method. Rich agents can invest more money than poor agents. If they make the same predictions, they will gain the same percentage return. However, due to the larger invested capital of the rich agent, this agent will earn much more in absolute terms and the gap between the two will become larger and larger.

This can be seen in Figure 3.3. This picture shows the growth of capitals of two strategies that both invested in the same index. It can be seen that when a capital becomes larger, its growth (but also its shrinkage) becomes faster and faster.

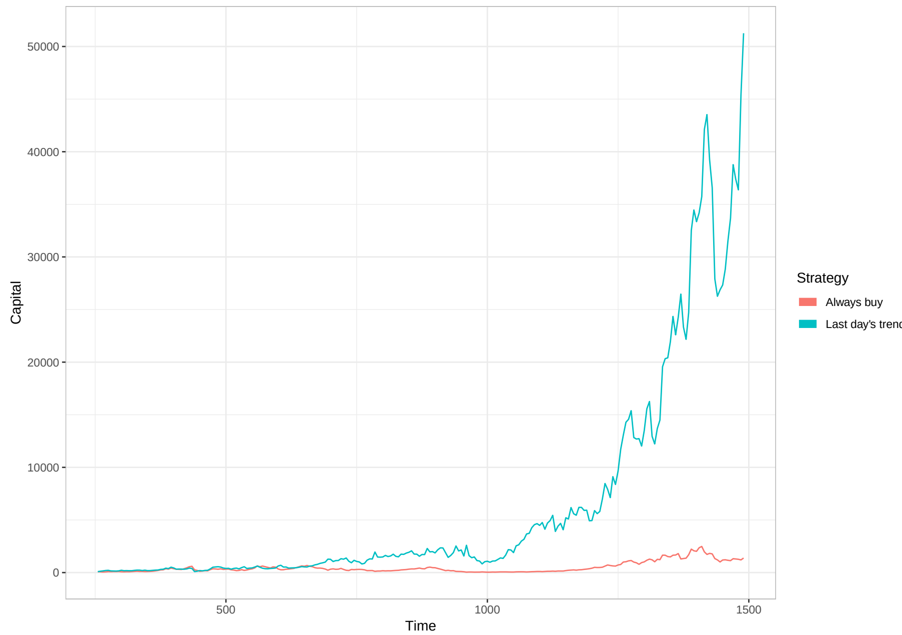

Figure 3.3: Growt of capitals using two strategies to predict the S&P500. Start capitals are equal for both strategies and ϵ is 0.1.

Figure 3.4 illustrates why this exponential growth is a problem. The left picture again shows the growth in capital. The right picture shows the mean return rate during the same period of time. It can be seen that between day 1100 and 1200, the always buy strategy (red) has on average a much higher return rate. However, as the capital of the last day's trend strategy (blue) is much larger, this capital still grows faster in absolute terms.

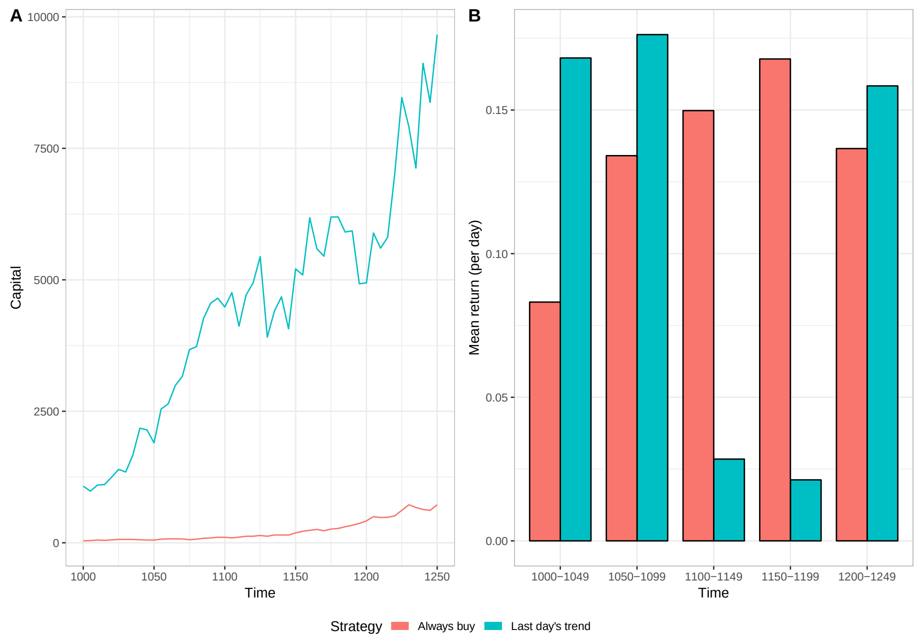

Figure 3.4: (A) The growth of capital between day 1000 and 1250. (B) The mean returns of both strategies during this same time period.

This is a problem because it makes the proportions highly dependent on the starting time and the starting capitals. An agent starting with a relatively high capital or growing very fast in the beginning, will hold this advantage during the entire simulation, due to the above given reason. Last, it seems unrealistic that people reinvest all their money in the same strategy. When capitals grow large, people may want to spread the risk and therefore spread their capital over several strategies. Also, people may want to switch to another strategy when the current strategy stays behind in performance, something that is not possible when solely using growing capitals. After all, even though the current strategy has given good results in the past, it may not be the best strategy in the future. Therefore, a more suitable method should be found to evolve the combination of agents in the population.

3.2.2 Replicator equation

A more promising method is to use the replicator equation. This is a very well-known method to update proportions of individuals in a population (Cressman & Tao, 2014). It is often used in both biology and game theory to simulate the evolution of either species or strategies. Its popularity is not in the last place due to its simplicity.

Interestingly, this method is also used to learn on real data and make predictions (Bolluyt & Comaniciu, 2018). Although the area of interest in the study of Bolluyt and Comaniciu (2018) differed from the one in the current study, the replicator equation has shown to be a useful method to make real-world predictions and therefore seems a promising method to use in the current study.

When used in game theory, agents and strategies are separated. The replicator dynamic is used to decide which proportion of the agents (or players) uses which strategy. It assumes that agents can switch between strategies, and therefore the proportions change over time. In this mechanism, the change in the proportion of each strategy is related to its success, or in biological terms its fitness, relative to the average fitness of the entire population of agents (Wieringa, 2001). This is captured in the following equation:

$$\Delta x_i = \alpha x_i * (f_i(x) - f(x)), \tag{3.4}$$

where x is a vector with the relative frequencies of all strategies, x_i is the relative frequency of strategy i, α is the adaptation speed and $f_i(x)$ and $f(x)$ represent the fitness of agents using strategy i and the average fitness of the population respectively. The fitness of an agent is generally either fixed or, through interaction, based on the other existing agents and their proportions of the population. However, in our study the fitness of an agent does not depend on its interaction with other agents, but on the trading success of this single agent.

To compute this fitness, the mean profit rate (MPR) (Liu & Wang, 2019) is used. This is shown to be a useful measure to evaluate the performance of investment models. The mean profit rate, as proposed by Liu and Wang (2019) is adapted in order to be able to handle the action of investing in a risk-free obligation as well. This resulted in the following very simple equation:

$$\text{MPR} = \frac{1}{n} \sum_{t=1}^{n} cr_t, \tag{3.5}$$

where n is the number of time steps on which the MPR is based and cr_t is the cognitive return rate the agent earned on time t. This raises the question what the cognitive return rate is and why this is used. That is explained in the subsequent paragraphs.

Depending on the action an agent chooses, and the index price on the next time step, the raw return rate is computed. When the agent invested in an index fund and predicted the right direction, it wins the logarithmic return on this fund. That is, when an agent bought the index fund and the price rose, or when it sold the fund when the price dropped, it earns $\ln(\frac{\text{new price}}{\text{old price}})$. When the agent invested in the index fund, but predicted the wrong direction, it looses this same rate. Furthermore, if the agent invested its money in the risk-free obligation, it will of course earn the (risk-free) logarithmic return rate on this obligation.

Logarithmic return rates are used because those are symmetric when they are continuously compounded. While simple arithmetic return rates are asymmetric. So if there are two consecutive changes in price, with arithmetic return rates it matters whether the price rises first and drops then, or the other way around. For logarithmic return rates this doesn't matter.

After computing the logarithmic return rate, transaction costs are subtracted if necessary. On the real stock market transaction costs are paid for every transaction one makes. We simulate this by subtracting transaction costs from the return rate when choosing another action than the day before. When choosing the same action, this is seen as holding a position and therefore no new transaction costs are charged. Those transaction costs are important, because when leaving them out, a strategy that switches between actions very often will possibly earn a lot of money, while in the real world its returns would be much lower because of the costs.

The logarithmic return rate lowered with the transaction costs that is now obtained has two disadvantages. First, the possible values are not limited from above. They can, in theory grow infinitely. Of course, those very high and low values are unlikely to occur, but still outliers can have an undesirably large influence on the proportions. Furthermore, using those logarithmic return rates does not represent real agents very well. Recall that the performance of models can be enhanced by fitting the real-world system more closely (Edmonds & Moss, 2001).

According to Dea, Gondhib, Manglac, and Pochirajud (2010) people are more influenced by the sign of previous returns than by the actual size of it, when making new decisions. Therefore, the cognitive return rate is used to represent peoples cognitive representation of its success better. This is calculated using the following equation.

$$cr_t = -0.5 + \frac{1}{1 + e^{-r_t}}, \tag{3.6}$$

where cr_t is the cognitive return rate on time t, and r_t is the logarithmic return rate on time t.

Using this cognitive return rate has two advantages. First, the cognitive return rate is limited between -0.5 and 0.5, which limits the fitness of agents in such a way that proportions cannot become negative as long as α is at most 1. Second, it is more similar to human cognition as the sign of the return has a larger impact on the value than the exact value of the logarithmic return rate. The further away from zero the logarithmic return rate is, the closer the values of the corresponding cognitive return rates are to each other.

As said, the return rates of multiple time steps are averaged to form the fitness of the agent. In that way, the relative frequencies of the agents are updated once every few time steps. Last, the average fitness over all agents is computed using the following equation:

$$f(x) = \sum_{i=1}^{N} x_i f_i(x), \tag{3.7}$$

where N is the number of different strategies or types of agents, x_i is the proportion of agent i and $f_i(x)$ is the fitness of agent i. By using the described replicator dynamic, the strategies that were relatively successful in the past weight relatively heavily in the next predictions, while strategies that were performing bad in the past weight relatively little.

All in all, the replicator equation allows agents to switch between strategies. This is an advantage over the method described in the previous section, where capitals are grown, because it allows people to put their money in a different strategy when the current one doesn't suffice any more. Furthermore, the replicator equation has been proven useful in multi-agent games. Additionally, it is used before to predict real data (although in another field), which is also our purpose. Combined, those arguments form the basis of the decision to choose this population dynamic for our model.

3.3 Strategies

As described before, there are many different investment strategies around (Keim & Madhavan, 1995). Those differ between very easy strategies, like "always buy the stock", and very complex strategies, like those based on deep neural networks (e.g. Yoshihara, Fujikawa, Seki, & Uehara, 2014). In this study, we choose to use simple strategies, as the focus is on the aggregation of the strategies and not on the strategies themselves. Another reason to choose simple strategies is because this makes it easy to understand the model and therefore, makes the results more easy to analyse and explain.

Furthermore, we choose to use strategies based on technical analysis instead of strategies based on fundamental analysis. The basis for this is two-fold. First, technical analysis is, on average, better suitable to predict short-term price trends, while fundamental analysis is more suitable to predict long-term price movements (Petrusheva & Jordanoski, 2016). As we try to predict movements one day ahead, strategies using technical analysis is a logical choice here. Moreover, technical strategies are based on historical price-data. As this data is unambiguous and easy to obtain, this can be preferred over the more elaborate and often less unambiguous data used by some fundamental strategies.

Ten different strategies are chosen to compare. Those strategies were developed in consultation with some experts in the area of investing from the pension fund PGB. Furthermore, there are also some strategies that are inspired by well-known strategies used in the area of game theory. Below, the different strategies are described.

Noise trading

Noise trading is a simple random trading strategy. At every point in time, the agent chooses a random action.

Always buying
This is a very simple strategy as well. It is based on the buy-and-hold strategy. This is a very simple, but relatively successful strategy where someone buys an index fund and holds it for a long time. In our model, agents do not hold index funds. They are assumed to sell it after every round to directly get their return. However, as the agents in our model do not need to pay transaction costs when performing the same action as the previous time step, the buy-and-hold strategy is comparable to a strategy in our model where the agent always buys. This strategy is implemented as the always buying strategy.

Always selling
This strategy is similar to the always buying strategy, with the difference that the agent is always selling instead of buying. As we assume agents do not hold index funds multiple rounds, agents do not have any funds to sell at the start of each round. Therefore, selling is interpreted as going short in our simulation, so that the agent can still profit from a fall in price

Risk-free investing
For completion, we also included a strategy where the agent always buys risk-free obligations. In this strategy, the agent does not invest in the index funds at all. It is important to include this strategy, because it may be possible that there are moments in time in which the risk of investing in the index fund is so high that this risk-free investment gives a higher return.

Trend following
Trend following strategies are very popular trading strategies. They are based on the assumption that current trends are likely to persist. There are lots of different trend following strategies. Here we use a simple moving average trading strategy, as described in Ellis and Parbery (2005). The simple moving average (SMA) is the average price of the index over a certain time window. This is formalized in Equation 3.8,

$$SMA = \frac{1}{n} \sum_{t=k-n+1}^{k} P_t, \tag{3.8}$$

where n is the number of time steps used in the average, k is the last time step used to calculate the average and P_t is the price of the index at time t.

When using the SMA as a trading strategy, k is generally set to the current time minus one time unit. In our case, the last observed closing price. When the current price lies at least one standard deviation above the moving average, the index fund is bought. Thereafter it is hold (in our model bought again without transactions costs) till the price drops below the moving average again. Opposite, when the current price lies at least one standard deviation below the moving average, the index fund is sold. This short position is then hold till the price rises above the moving average again. In all other situations, there is invested in the risk-free obligation.

Mean reverting

A similar type of strategy that is used is the mean reverting strategy. This one is exactly opposite to the trend following strategy. When the current price is at least one standard deviation above the moving average, it is expected to fall, so index funds are sold. If, on the other hand, the current index price is at least one standard deviation below the simple moving average, the index fund is bought (Stasinakis & Sermpinis, 2014). Those positions are again hold till the price hits the moving average. Otherwise, the agent chooses to invest in the risk-free obligation.

The underlying assumption of this strategy is that an asset is over- or undervalued when it's price rises above or falls under the moving average. As the asset is expected to return to its average value, overvalued assets are expected to fall in price, while undervalued assets are expected to rise in price.

Last day's trend

This strategy is a very easy strategy based on the strategy Tit-for-Tat. This is a well-known strategy from game-theory where a player starts cooperating and thereafter always replicates the opponent's previous action. Of course, we do not have different agents playing against each other, but separate from each other, so this strategy cannot be used directly. Therefore, we adapted the strategy a bit, such that the agent starts with buying and then perform the action that would have been best on the previous day. So if the market price rose last day, the player will buy, and when it fell, the player will sell.

Last two day's trends

Instead of using Tit-for-Tat as an inspiration, this strategy uses Tit-for-two-Tats as inspiration. That strategy is similar to Tit-for-Tat with the difference that it only stops cooperating when the opponent does not cooperate for two turns in a row. When the opponent thereafter cooperates again, the player will also cooperate till the opponent stops cooperating for two subsequent turns again.

In our adaptation of this strategy, the agent always buys, except when the price fell both last two days. If that is the case, the player sells. This may seem a bit strange on first sight, as the agent will buy much more often than it will sell. However, on average, stock prices grow more often than they fall, so buying may be the savest option most of the time.

Win or change

The win or change strategy is based on the game theory strategy of Pavlov. When following this strategy, the action of the previous turn is repeated when it was successful. When the last action was not successful, another action is chosen. This can easily be transformed into a trading strategy. When the action of the previous day earned the highest possible reward (remember that there are two rewards possible: the risk-free reward and the reward of the S&P500), this same action is chosen today. Otherwise, a random action is chosen.

Majority

The last trading strategy that is used is the majority strategy. This is a very easy strategy that counts how often buying, respectively selling, would have been the best action in the previous n days. When selling has most frequently been the best action, selling is chosen as new action. In the same way, buying is chosen when that has been the best action most often in the past days. When both selling and buying were equally successful during the past days, the risk-free investment is chosen. The time window which the strategy evaluates is variable.

4. First experiment

In this experiment, it is tested whether a multi-agent model with dynamically updating proportions, using the replicator equation, can be used to predict stock prices. To answer this question, the performance of such a model is tested with multiple different conditions and is compared to other models.

4.1 Methods

For this experiment, we use the model as defined in Chapter 3. We test its one day ahead prediction performance on the S&P500, which is the index of the 500 largest companies in the United States. It may be good to keep in mind that the methods described for this experiment will also be used for the second experiment.

The data we use to train and test the model contains the daily closing prices of the S&P500 from 1955 up to and including 2018. Besides the historical closing prices of this index, the agents also observe the daily risk-free return on the USA treasury bills, as a measure of the returns on a risk-free investment.

Initially, all types of agents cover an equal part of the population. This is done to make sure differences in proportions are based on differences in the success of strategies and not on random initial differences. Thereafter, the replicator equation is used, as described in the previous chapter, to update the proportions of the different types of agents. The first part of the data is used to train the model. As the initial proportions are set arbitrary, without looking at the data, the model cannot be expected to make good predictions from the start on. Therefore, the first data points are used to come to a more useful set of proportions.

After this short training phase, we test the performance of the model. To do this, the model is ran over several years of data. Every day, it is checked whether the model made a good prediction or not by comparing the predicted trend direction to the real trend direction.

4.1.1 Performance metrics

To measure how good the predictions of our model are quantitative performance metrics are used. We use three different performance metrics that are used in economics. Those are described below.

Hit rate
The hit rate is a very simple performance metric. It represents the percentage of the time that the model predicts the direction of the change in price correctly (Grothmann, 2003). This is formalized in Equation 4.1,

$$\text{hit rate} = \frac{\text{nr of correct predictions}}{\text{total nr of predictions}}. \tag{4.1}$$

Accumulated return
The accumulated return, represents the total return the model would have earned if it had traded based on its predictions (Grothmann, 2003). This measure adds the return rates the model earned over the whole time period. As with the performance of the agents, transaction costs are subtracted when the model performs another action than the previous day. Furthermore, again the logarithmic returns are used. The accumulated return is used to study the models effectiveness in earning money. The exact computation is shown in Equation 4.2.

$$\text{Accumulated return} = \sum_{t=0}^{T} (r_t - c_t), \tag{4.2}$$

where r_t is the logarithmic return as a percentage of the invested money on time t. Transaction costs are subtracted from the return. The costs are a percentage of the investment and are represented by c_t. The total number of predicted time steps is represented by T.

Sharpe ratio
The last performance metric is the Sharpe ratio. This is a popular metric in economics, as it does not only take returns, but also risk into account. It is defined as the excess return divided by the standard deviation of this excess return (Sharpe, 1994). The excess return is the return of the model (as a percentage) minus the risk-free return rate (using USA treasury bills). The whole equation is shown in Equation 4.3.

$$\text{Sharpe ratio} = \frac{\text{mean}(R - Rf)}{\text{sd}(R - Rf)}, \tag{4.3}$$

where R is the vector with all model returns during a certain period of time and Rf is a vector with the risk-free returns during that same time period. The value obtained is the Sharpe ratio per day.

4.1.2 Optimization

To make accurate predictions, appropriate values for the learning rate (α in Equation 3.4) and the update frequency (n in Equation 3.5) should be used.

The update frequency n represents the number of days that are combined into a mean profit rate. After this n days, the proportions are updated and stay fixed for the next n days. The learning rate, α, indicates how fast the proportions are updated.

To find out which values for n and α are appropriate, the performance of the model is systematically tested with different values. The update frequency can range from every day to never. However, as our model is build upon the assumption that the dynamic characteristic of the proportions is beneficial, it should be updated often enough. Therefore, the different values we tried are 1, 5 and 10 days. That is, every day, week, or two weeks.

Alpha can range between 0 (the proportions do not change at all) and infinity (the new proportion only depends on the difference between individual and population fitness). However, those extreme values are not interesting for our current purpose. We are more interested in a model where both the previous proportions and the current fitness are represented in the new proportions. Therefore, we use the more moderate values 0.5, 1, and 2.

To test which of those are the most appropriate, we run our model multiple times to see which combination of α and n give the best performance as measured by the three performance metrics described before. We use three different time frames to test our model on. In that way, we make sure that the values we get are not accidentally optimal at the single time serie we test the model, but are optimal over multiple different time ranges.

We run the model on the S&P500 with the parameter settings as shown in Table 4.1. As those parameters are no key parameters in our model, they are set to some predefined, but logical values. The time ranges we use are 1955 to 1960, 1975 to 1980 and 1995 to 2000. On all those time frames, the model is ran 5 times with all different possible combinations of n and α. This resulted in 135 runs in total.

Parameter	Value
Transaction costs	0.1 percent
Training length	126 days
Number of strategies	10 (all described strategies)
Time window mean reverting strategy	20 days
Time window trend following strategy	20 days
Time window majority strategy	10 days

Table 4.1: Parameter settings

The three performance metrics described in the previous section are computed on those 135 runs. Thereafter, the results on the different runs and different time windows are combined into one value for each combination of α and n. The results are shown in table 4.2 and 4.3. Table 4.2 shows the means on all three performance metrics. The first two columns show the different values that are tested of α and n respectively. The next three columns represent the mean accumulated return, hit rate and Sharpe ratio over the fifteen simulations that are ran with the respective learning rate and update frequency in that row. It can be seen that the combination of a learning rate of 1 and an update frequency of 1 gives the highest performance on all three measures. This row is highlighted in green.

learning rate	update frequency	accumulated return	hit rate	Sharpe ratio
0.5	1	93.9	0.550	0.087
1.0	1	95.8	0.550	0.089
2.0	1	57.5	0.498	0.039
0.5	5	73.4	0.538	0.067
1.0	5	90.2	0.546	0.084
2.0	5	94.8	0.550	0.088
0.5	10	76.7	0.536	0.069
1.0	10	76.1	0.538	0.068
2.0	10	89.4	0.546	0.082

Table 4.2: Mean performance

Table 4.3, on the other hand, shows the standard deviations over the different runs and time windows. A lower standard deviation is preferable, because this indicated that the performance of the model is more stable. However, the standard deviation on the combination α is 1 and n is 1 (the red row) is quite high, especially on the accumulated return. This standard deviation is much lower when using a learning rate of 1, but an update frequency of 5 (the green row). As the mean performance when using those values is still high, this last combination of α and n is chosen.

learning rate	update frequency	accumulated return	hit rate	Sharpe ratio
0.5	1	21.7	0.001	0.022
1.0	1	19.8	0.003	0.021
2.0	1	55.5	0.081	0.068
0.5	5	17.3	0.010	0.025
1.0	5	13.5	0.006	0.020
2.0	5	20.2	0.001	0.022
0.5	10	14.0	0.012	0.019
1.0	10	15.6	0.011	0.021
2.0	10	13.4	0.005	0.018

Table 4.3: Standard deviation of the performance

4.1.3 Performance evaluation

Now the optimal values for n and α are found, the prediction performance of the model can be tested. Again, the model is used to make one-day-ahead trend predictions on the S&P500. Three different time windows of 15 years each are used to test the performance of the model on. Those time windows, even as the other parameter settings, are shown in Table 4.4.

Parameter	Value
Learning rate	1
Update frequency	5 days
Transaction costs	0.1 percent
Training length	126 days
Number of strategies	10 (all described strategies)
Time window mean reverting strategy	20 days
Time window trend following strategy	20 days
Time window majority strategy	10 days
Test period 1	1960-1975
Test period 2	1980-1995
Test period 3	2000-2015

Table 4.4: Parameter settings

With those parameter settings, the model is ran 50 times on all three time windows. This gives 150 series of predictions in total. On all those 150 series of predictions the three performance metrics are computed. That is, the hit rate, the accumulated return and the Sharpe ratio are calculated. Thereafter, the models mean performance is compared to that of other forecasting strategies. This is done step-by-step. Starting with a comparison with an easy strategy and comparing it to better and better strategies till the model can not outperform the strategy it is compared to any more. The different comparisons that are made are described below.

Random prediction

First, the performance of the model is compared to that of some of the strategies that our model combines, starting with the random strategy. The three performance metrics are computed over the 150 runs for the random strategy as well. With a multivariate repeated measures ANOVA it is tested whether the model performs, on average over the 150 different runs, significantly better than this random strategy. An ANOVA is used, because it is an omnibus test that makes it possible to analyse the performance on all three time periods together. When the model does outperform the random strategy, it is tested whether this holds true over all three time periods separately. This is tested by running nine different paired sample t-tests (one for each combination of performance metric and time period). The p-values are corrected for multiple comparisons using a Bonferroni correction.

Buy and hold

When the model is proven to be better than the random strategy, a slightly harder test is conducted. That is, the performance of the model is tested against the performance of the buy and hold strategy. As there is no random factor in this strategy, its performance over the 50 runs of one time window will be equal. Therefore, several one sample t-tests are used to test the models performance against the buy and hold strategy instead of an ANOVA and paired t-tests. Nine different one sample t-tests are used, one for each combination of time period and performance measure. In those t-tests, it is tested whether the performance of the model is on average significantly higher than the performance of the buy and hold strategy. Again, Bonferroni correction for multiple comparisons is applied. This time, no omnibus test is used to test the difference in performance over all three time periods combined. There is, to the best of our knowledge, no test to do this without violation of the assumptions of the test.

Best performing strategy

An even harder test is to compare the models results with that of the best of the strategies it combines. Of course, there are multiple methods possible to choose what the best strategy is. It is chosen to choose a best strategy for the three time periods separately. As there is a gap of at least 25 years between the first and the last period, it seemed unfair to choose one best strategy for all three time periods.

Therefore, we choose the strategy performing the best (of the 10 strategies the model combines) during the tested 15 years as the best strategy. Of course, this is looking backwards to find the best strategy. This would never have been possible in real time situations, as you have to choose your strategy before trading. However, as a performance test it is useful to compare the performance of the model to the strategy that performed the best, not to the strategy that was predicted to perform the best. After all, it is useful to see whether the model can outperform all strategies it is composed of.

Depending on the strategy that is chosen (whether this has a random component or not), paired-sample t-tests or one-sample t-tests are used to test whether the model performs significantly better than the best strategy. If possible, the main effect over the three time periods together is tested first. If this is significant or impossible to test, simple effects for each time period and performance measure separately are tested as well. Of course, Bonferonni correction is applied.

Equal weights

When our model can outperform the best of the models it is composed of, the hardest test is conducted. That is, the model is tested against a combination model with equal weights. This model combines the same strategies as our dynamic model does. As discussed in chapter two, equal weight models are high performing combination models. However, a condition to perform well, is that all combined prediction models perform approximately equally well. Therefore, trimming is used to remove the 50% worst performing strategies, and only the best performing once are used.

The obtained equal weight model is ran 150 times as well (50 times for each time period), and its mean performance is compared to that of our dynamic model. This is a useful test, as is shows whether the dynamic characteristic of our model gives it an advantage over stable models or not. The method to compare the performance is equal to the method used to compare the mean performance of the random strategy.

Robustness
When the model is shown to perform good, that is, when it can outperform all four strategies described above, it is important to check the robustness of this performance. Does it only perform good with the specific settings on our test sets, due to some good luck? Or does the model perform well over multiple test sets and settings. To check its robustness, two different tests are performed.

First, the influence of the initial composition of the population is tested. To do this, the model is ran with random initial proportions for each strategy. It is tested whether the performance stays stable over those different initializations, or whether it depends a lot on the initial proportions.

Second, the influence of the starting time is tested. To do this, the model is ran over the last ten years of each 15 year time periods only. So it does exactly the same as our original model, only starting five years later. The performance of both the original model and the later-starting model over those last 10 years is calculated. It is then tested whether this performance is different for the 15 year model compared to the 10 year model. Again, an ANOVA and paired t-tests are used.

4.2 Results

The results of the performed analyses are discussed in this section. As said, our model is ran over three different time periods. The prediction performance is summarized in Table 4.5. The first three rows show the average hit rate, accumulated return and Sharpe ratio over the 50 runs of one time period. The last row shows the average over all 150 runs.

Time	Hit rate	Accumulated return	Sharpe ratio
1	0.60	535.58	0.18
2	0.53	197.16	0.03
3	0.49	-47.34	-0.01
Mean	0.54	228.46	0.07

Table 4.5: Prediction performance of our model

It can be seen that the performance of the model differs across time periods. In the first period, it does quite well, with a hit rate of 60% and an accumulated return of over 500%. However, the last period, the model does much worse. It even looses money! Indeed, a multivariate one-way ANOVA pointed out that the mean performance of the model differs significantly across the three time periods.

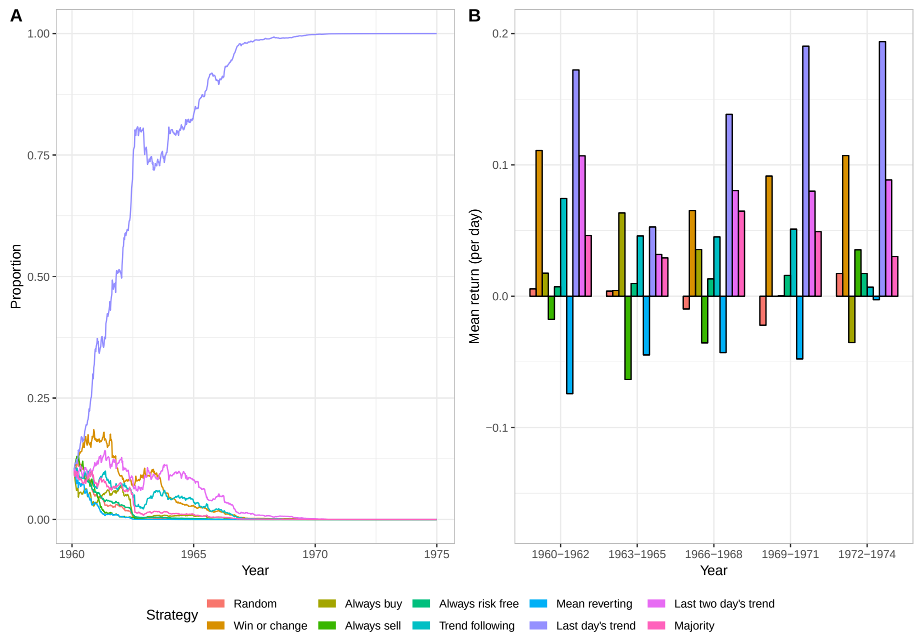

Figure 4.1: (A) The dynamics of the proportions of the strategies between 1960 and 1975. (B) The mean returns of all strategies during this same time period.

These time dependencies may seem problematic, but part of it is easy to explain. Figures 4.1B, 4.2B and 4.3B show the mean returns of the individual strategies during the first, second and third time period respectively. It can be seen that the mean returns are very high during the first period (especially for the best strategy), lower for the second period, and lots of returns are even negative during the last period. Therefore, the time dependence of the performance is at least partly caused by the changed market structure.

Another observation that can be made from Figure 4.1 to 4.3 is that not just the performances, but also the proportions of the strategies differ over time. During the first period, there is one strategy that makes up almost the entire population. In later periods, the population is more diversified. However, there are still many time periods were one strategy makes up at least 50% of the population. As the performance is so time dependent, it is important to compare the performance if the model with that of other models. This is done as described in the previous section. The results are discussed below.

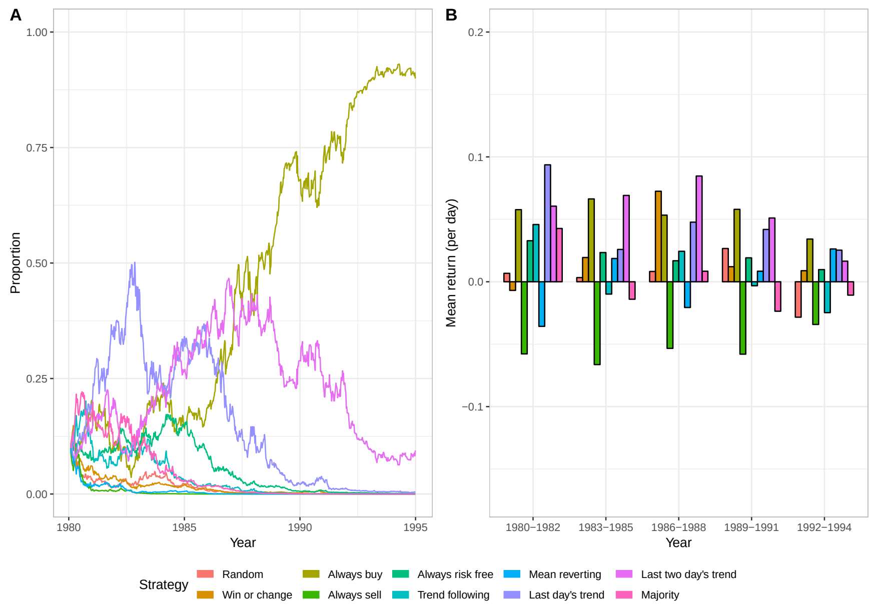

Figure 4.2: (A) The dynamics of the proportions of the strategies between 1980 and 1995. (B) The mean returns of all strategies during this same time period.

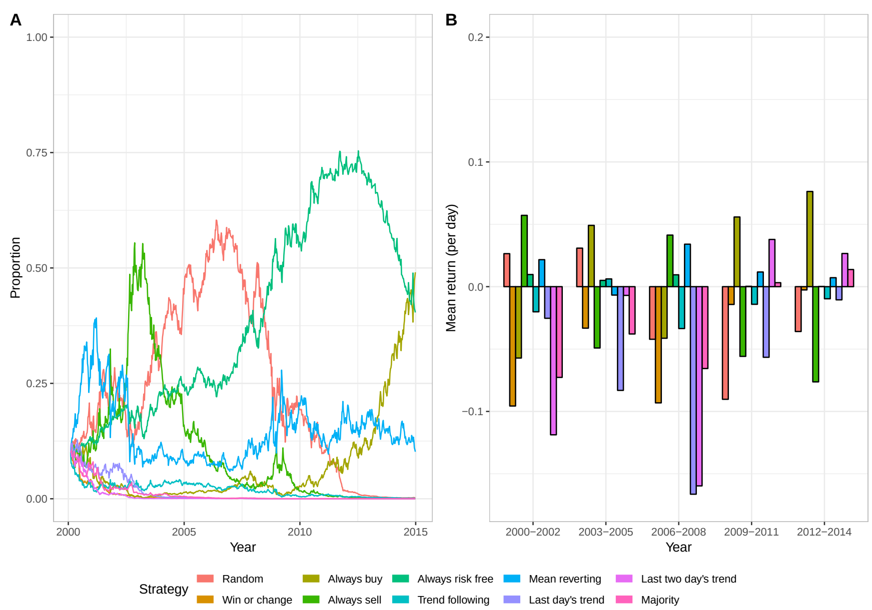

Figure 4.3: (A) The dynamics of the proportions of the strategies between 2000 and 2015. (B) The mean returns of all strategies during this same time period.

Random prediction
First, the performance of our model is compared against a random forecasting strategy. The results of this strategy are shown in Table 4.6. The difference in performance between our model and the random model is shown in Table 4.7. It can be seen that our model performs on average significantly better than the random model over the entire 45 tested years on all three performance measures.

Post-hoc comparisons showed that the mean performance of our model is significantly higher than the mean performance of the random model on the first and second time period. However, the average accumulated return of the random model is significantly higher than that of our model during the third time period. All in all, we can conclude that our model performs better than the random model, on average, but that this advantage is not stable over time.

Time	Hit rate	Accumulated return	Sharpe ratio
1	0.33	-6.98	-0.02
2	0.33	0.72	-0.02
3	0.33	-17.04	-0.01
Mean	0.33	-7.77	-0.02

Table 4.6: Performance of a random prediction strategy

Time	Hit rate	Accumulated return	Sharpe ratio
1	0.26*	542.55*	0.21*
2	0.19*	196.44*	0.06*
3	0.16*	-30.30*	-0.005
Mean	0.21*	236.23*	0.09*

Table 4.7: Difference in performance between our dynamic model and a random strategy. Significant differences ($p < 0.05$) are marked with a *.

Buy and hold
The buy-and-hold strategy is expected to be better than the random strategy, and therefore, harder to outperform. In Table 4.8 it can be seen that this is indeed the case. Table 4.9 shows the difference in performance between this strategy and our model. It can be seen that the differences are smaller than those with the random strategy.

Again, our model performs on average better than the buy-and-hold strategy. However, further analysis shows that the relative success of our model compared to the buy-and-hold strategy differs between the different time periods. On the first time period, our model performs significantly better. However, on the second time period, the differences between the two methods are very small. Although our dynamic model has a significantly larger return and Sharpe ratio, the hit rate is lower. On the third time period, our model even performs significantly worse on all three performance metrics than the buy-and-hold strategy. It can be concluded that our model performs better than the buy-and-hold strategy on average, but this is highly time dependent.

Time	Hit rate	Accumulated return	Sharpe ratio
1	0.53	63.01	0.01
2	0.53	192.47	0.03
3	0.54	62.27	0.01
Mean	0.54	105.92	0.02

Table 4.8: Performance of a buy-and-hold strategy

Time	Hit rate	Accumulated return	Sharpe ratio
1	0.06*	472.56*	0.18*
2	-0.003*	4.70*	0.001*
3	-0.05*	-109.61*	-0.02*
Mean	0.003^{Δ}	122.55^{Δ}	0.05^{Δ}

Table 4.9: Difference in performance between our dynamic model and a buy-and-hold strategy. Significant differences ($p < 0.05$) are marked with a *, differences that could not be tested on significance are market with a $^{\Delta}$.

Best performing strategy

The third test we conducted is to compare the combined model to the best performing one of the strategies it combines. The strategies with the maximum score on the different performance metrics for each time period are summarized in Table 4.10. For each time period, we chose the strategy that has the best performance on the most metrics as the best one. That is "last day's trend", "last two day's trends" and "always buy" for the first, second and third time period respectively.

Time	Hit rate	Accumulated return	Sharpe ratio
1	Last day's trend	Last day's trend	Last day's trend
2	Always buy	Last two day's trends	Last two day's trends
3	Always buy	Last two day's trends	Always buy

Table 4.10: Best performing strategy

The performances of those strategies are summarized in Table 4.11. The first row represents the performance of the last day's trend strategy during the first time period, the second row the performance of the last two day's trends strategies during the second time period and third row the performance of the always buy, or buy-and-hold, strategy in the third period. The last row shows the mean performance of those strategies on the three time periods.

Table 4.12 shows the difference between the mean performance of our model and the best strategies. It can be seen that our model performs, on average,

Time	Hit rate	Accumulated return	Sharpe ratio
1	0.60	537.43	0.19
2	0.52	206.99	0.04
3	0.54	62.27	0.01
Mean	0.55	268.90	0.08

Table 4.11: Performance of the best performing strategies

Time	Hit rate	Accumulated return	Sharpe ratio
1	-0.000	-1.85*	-0.001*
2	0.011*	-9.83*	-0.003*
3	-0.052*	-109.61*	-0.024*
Mean	-0.014$^{\Delta}$	-40.43$^{\Delta}$	-0.009$^{\Delta}$

Table 4.12: Difference in performance between our dynamic model and the best performing strategies. Significant differences ($p < 0.05$) are marked with a *, differences that could not be tested on significance are market with a $^{\Delta}$.

significantly worse than the best of the strategies it combines. There are two exceptions. First the average hit rate does not differ significantly between our model and the last day's trend strategy during the first time period. Second, the hit rate of our model is significantly higher than that of the last two day's trends strategies during the second time period.

However, all in all it can be concluded that our model cannot outperform the best of the models it combines. We could test if this lack of prediction success is stable over different initializations of time and proportions, and whether our model performs worse than the equal weights model. But as this equal weights model should be even better than the best performing strategy it is composed of, both would be useless. Therefore, no further comparisons are made.

4.3 Discussion

In this experiment it was tested whether a multi-agent model that uses the replicator equation to update the composition of agents can be used to predict trends of stock index prices. It turned out that although it can make good predictions and earn money, the model is not fully suitable for this task.

First of all, its performance is highly time-dependent. Although the model is, on average, better than a random prediction strategy or the buy-and-hold strategy, there are times on which it is even worse than the random strategy. Of course, this is highly undesirable.

Furthermore, the model cannot outperform the best of the strategies it is composed of. Therewith, it does not give an advantage over using one single strategy. On the first tested time period, this is not surprising. There is one strategy that is, over the whole time period much better than the others. Therefore, it makes sense that the model only uses this strategy to base it's predictions on and performs approximately equal to this best strategy.

However, during the second and third time period, there is no single best strategy that outperforms all others during the entire 15 year period. Therefore, it would be best to switch between strategies or combine strategies during those time periods. Indeed, Figures 4.2A and 4.3A show that strategies are combined more often during those periods. The strategy that has the highest proportion changes over time as well.

Even though the expected switching and combining of strategies is present, the model performs bad on the last two time periods. There are multiple reasons for this. First, the returns of the individual strategies are much lower in the second time period than in the first one. Most strategies even loose money in the third period. This explains the lower returns of our dynamic model, but does not explain why it cannot outperform the strategies it combines.

This can be explained by the slow adaptation of the model. The proportions of the strategies are updated based on their previous trading success. Therefore, the model lags behind per definition. A strategy that was good in the past, after all, is not guaranteed to be good in the future as well. This could be prevented by updating the proportions based on the strategies predicted success instead of its past success.

Furthermore, the proportions change very slow. This is, for example, visible in Figure 4.3. The random strategy (the red one) looses money from 2006 onwards. However, it takes to 2011 before it makes up less than 10 percent of the population. Similarly, the always buy strategy (the yellow one) performs very good between 2003 and 2005. However, it is not able to grow fast enough to benefit from this. Later, it starts performing well from 2009 onwards, but in 2012 it still makes up only 10 percent of the population.

All in all, proportions change too slow to keep up with the rapidly changing market. By the time the proportions adjusted, the market is already changing again and many possible profits have passed. This problem is mainly caused by the fact that the fitness of agents is based on external data, instead of interactions as in most applications of the replicator dynamics. When using the interaction between agents, the change in the fitness depends on the change in the proportions. Therefore, the change in fitness is more stable and predictable than when using external data. This higher stability makes it possible for the proportions to adapt to the new environment. In our application, on the other hand, the market structure (that sets the prices) can change faster than the proportions.

A possible solution would be to use a higher learning rate, so that proportions update faster. However, the stock market is very volatile. Therefore, the performance of a strategy can vary a lot between days. When the learning rate is too high, the proportions would depend too much on accidental success instead of structural success.

Another possibility would be to use adaptive learning rates. Those can be high when fast updating is needed, but low when more stable proportions are desirable. The possibilities of such an adaptive learning rate are explored in the next experiment.

5. Second experiment

In the first experiment, it is shown that a multi-agent model with dynamically updating proportions, using the replicator equation, cannot outperform the best of the individual strategies it combines. One problem with this model is the slow updating of proportions compared to the rapid changing financial market. In this study, a revised version of the model is proposed with a faster adaptation. It is tested whether this revised model performs better than the original model in the previous experiment.

5.1 Model

For this experiment we use, again, the dynamic multi-agent model as defined in Chapter 3. However, we make a small adjustment to make it update faster. The most easy adjustment would, of course, be to simply increase the learning rate. That is, to increase α from Equation 3.4. However, this would make the model highly dependent on accidental success or failure.

Consider, for example a random strategy. On average, this strategy would not perform very well. However, it will by change perform well at some days. When the learning rate is too high the strategy will grow very fast, even though its performance was just caused by good luck instead of a systematic high performance. As this is undesirable, some other solution should be found.

One possibility is to use a variable learning rate. When there is one stable, best strategy the model should learn slow. This would prevent lucky strategies to become too large, or unlucky, but good strategies to fall too fast. This would, for example, be desirable in the period between 1960 and 1975. As seen in Figure 4.1 there is one best strategy during this time period. It is desirable that this strategy makes up a large part of the population over the entire time range.

In the period between 2000 and 2015, on the other hand, the performance of strategies differ a lot over time. Strategies that are very good at the beginning are much worse later on and the other way around. Therefore, at this period it would be desirable to have a higher learning rate to be able to adjust quickly to the underlying changes in the financial market.

A possibility would be to make the learning rate dependent on the volatility of the market. However, it is not guaranteed that the performance of strategies is less stable in volatile markets than in more stable ones. For example the risk-free investment could be the best strategy over a long period in time in a volatile market.

Another possibility is to let the learning rate depend on the success of the model as a whole. That is, to look at the performance of the prediction of the combination of all strategies. When the model performs very good, it should not change too much. When it performs bad, on the other hand, the learning rate should increase so that the model can learn a better combination of strategies.

Although this would definitely be possible, it assumes that all agents collaborate to find the optimal combination of strategies. This is a very unlikely assumption. When one strategy is performing well, agents using this strategy are happy. Why should they switch strategies just because the population as a whole is losing money? Similarly the other way around. When agents want to switch to another strategy (for example because they are losing money), they would not be stopped because the population as a whole is performing well.

It is more appropriate to assume that agents decide how fast they want to switch between strategies on their own. This can be done using an individual learning rate. In that way, not all agents are forced to switch with the same rate. Such an individual learning rate is used before in the ReDVaLeR algorithm of Banerjee and Peng (2004). This algorithm is explained in the next section.

ReDVaLeR

ReDVaLeR is an abbreviation of Replicator Dynamics with a Variable Learning Rate (Banerjee & Peng, 2004). It is a WoLF-like adaptation of the replicator equation. WoLF stands for win or learn fast. The idea behind this is that agents that are successful learn slower than agents that are less successful. In Equation 5.1 this WoLF-principle is combined with the Replicator Dynamic.

$$\Delta x_i = \alpha x_i * (\eta_i f_i(x) - \sum_{i=1}^{N} x_i \eta_i f_i(x)), \tag{5.1}$$

where x_i is the proportion of strategy i in the population, α is the learning rate, η_i is the individual learning rate of strategy i, $f_i(x)$ is the fitness of strategy i and N is the number of different strategies.

In the original description of the ReDVaLeR algorithm, the individual learning rate depends on the Nash equilibrium. If the payoff of a strategy is less than that of the Nash equilibrium, the individual learning rate is higher than 1. If the payoff of a strategy is higher than or equal to that earned in a Nash equilibrium, the individual learning rate is lower than 1.

Adjustment on ReDVaLeR

The ReDVaLeR as proposed by Banerjee and Peng (2004) cannot directly be applied to our model. As we use external price data, there are no fixed Nash equilibria which can be used to find the individual learning rate. Furthermore, WoLF does not seem to be the best algorithm for our purpose. When strategies do very well, but are very small, we would want them to learn fast. However, as they are doing well, they would not learn fast when using WoLF.

Therefore, another method to find the individual learning rate is proposed in this study. The assumption underlying this method is that agents learn fast when the performance goes against the expectations and learn slow when the performance meets the expectations.

First, the expectation and performance should be defined. The expectation is based on the proportion a strategy has. A strategy with a large proportion is expected to predict the trend better than a strategy with a small proportion. Therefore, the proportion of a strategy is seen as a indicator of the expected success of this strategy.

To measure performance, the mean profit rate (MPR), as described in Equation 3.5, is used. However, even the best strategy has some days on which it looses money. It is undesirable that its learning rate changes too fast due to some accidental returns. Therefore, a long-term performance is calculated. That is, the MPR over the last month is used as a measure of the performance of a strategy. The assumption behind this is that agents only start learning faster when a strategy is performing differently than the expectations for a longer period of time.

Now we have measures for the expectation and the performance, but they are not directly comparable. There are multiple possible ways to compare them to each other. In this study, rankings are used. The strategy with the largest proportion is expected to have the highest performance. The strategy with the smallest proportion, on the other hand, is expected to have the lowest performance.

Both the proportions and the MPRs of all strategies are ranked from low to high, giving the numbers one up to the number of strategies. Thereafter, the difference between the two rankings is calculated for each strategy. Strategies that perform as expected will have (approximately) similar rankings for both the proportion and the MPR. For strategies that perform differently than expected, the difference between the two rankings will be larger.

Now, we have a value for the difference between the expectation and the performance. The only task left is to transform this value into a learning rate. By dividing the value by the number of different strategies minus one, we get a value between 0 and 1. As our aim is to increase the adaptation speed of the model when it is performing bad, the learning rate should be able to become larger than 1. Also, a learning rate of zero would be a bit too low. Therefore, a constant is added to the obtained value. With this constant, it can be set when the learning rate is lower or higher than 1. This entire calculation is summarized in Equation 5.2.

$$\eta_i = \frac{|\mathrm{RankMRP}_i - \mathrm{RankProportion}_i|}{N-1} + C, \tag{5.2}$$

where η_i is the individual learning rate, $\mathrm{RankMRP}_i$ is the ranking of strategy i on the MPR and $\mathrm{RankProportion}_i$ is the ranking of the proportion of that same strategy. As before, N represents the total number of strategies and C represents a constant.

So, all in all, our adjusted model is similar to the model used in first experiment. The only difference is that individual learning rates are added to the model. That is, Equation 3.4 is replaced by Equation 5.1. The individual learning rates are calculated using Equation 5.2. As the new proportions obtained by using Equation 5.1 are no longer guaranteed to be positive, the new proportions are rescaled to be positive. The minimum proportion is set to a very small, but positive value. The performance of this adjusted model is tested in the same way as the original model in the first experiment. The results are discussed in the next section.

5.2 Results

The same tests are performed as in the first experiment. That is, the revised model is used to predict the one-day-ahead trend of the S&P500. The same model parameters are used as in the previous experiment. Those are shown in Table 4.4. The constant that is added to the individual learning rate in Equation 5.2 is set to 0.5. This makes the individual learning rate range from 0.5 to 1.5.

Again, 50 runs per time period are used. The evolution of the proportions of the strategies on one of those runs is shown in Figure 5.1 for the second time period. The proportions when using the new model are shown at the right. As a comparison, the proportions when using the model of experiment 1 are shown at the left.

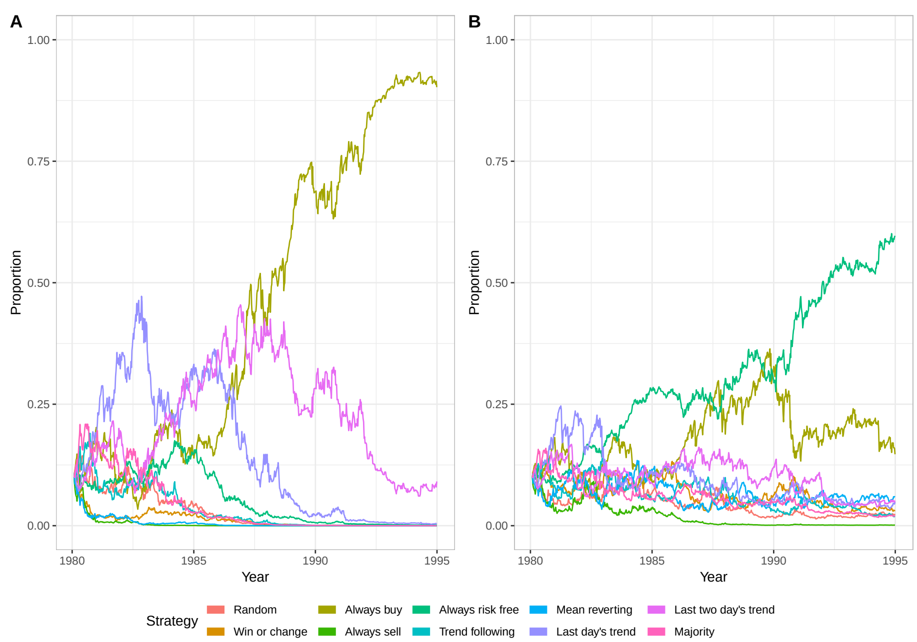

Figure 5.1: (A) The dynamics of the proportions of the strategies between 1980 and 1995 when using the model of experiment 1. (B) The same dyamics when using the model of experiment 2.

It can be seen that in the revised model, the proportions evolve differently than in the original one. At some times, the changes are slower in the modified model. Look, for example, at the period just before 1990. The always buy strategy (yellow) grows much faster in the first experiment. A bit later, however, the opposite is true. Just after 1990, the proportion of this same strategy is dropping. It can be seen that it drops faster in the second experiment than in the first experiment. This results in the total prediction of the model being more of a combination of strategies, instead of a combination of only the best one or two strategies as in the original model. What the influence of this is on the performance of the model is discussed below.

The mean performance of the model is shown in Table 5.1. It can be seen that the performance differs over time. The models performance is compared to other investment strategies in the same way as described in Section 4.1.3. That is, first the model is compared to a random strategy, the buy-and-hold strategy and the best of the strategies it combined. When performing well, it is tested against an equal-weight strategy and its robustness is tested. The results are described below.

Time	Hit rate	Accumulated return	Sharpe ratio
1	0.60	533.69	0.18
2	0.52	180.61	0.03
3	0.52	-96.90	-0.02
Mean	0.55	205.80	0.06

Table 5.1: Mean performance of the model, for each of the three periods separately and in the last row for all three periods together.

Random prediction

The performance of our model is compared against a random forecasting strategy. Table 5.2 shows the average performance of this strategy. In table 5.3, the difference in performance between our model and the random model is shown. The random strategy performs significantly worse than our model on average over all three time periods. The random model also performs, on average, significantly worse than our model during the first two time periods. However, between 2000 and 2015 (period 3), the random model outperforms ours on two of the three performance measures.

Buy and hold

Next, it is tested whether our model can outperform a simple buy and hold strategy. The performance of the buy and hold strategy is already shown in Table 4.8. This is the same as in the previous experiment, as this strategy has

Time	Hit rate	Accumulated return	Sharpe ratio
1	0.33	-14.55	-0.03
2	0.33	-1.09	-0.02
3	0.33	-19.87	-0.01
Mean	0.33	-11.83	-0.02

Table 5.2: Mean performance of a random trading strategy.

Time	Hit rate	Accumulated return	Sharpe ratio
1	0.26*	548.24*	0.21*
2	0.19*	181.70*	0.05*
3	0.18*	-77.04*	-0.01*
Mean	0.21*	217.63*	0.08*

Table 5.3: Difference between the mean performance of our model and the random trading strategy. A * indicates that the difference is significant ($p < 5$).

no random component. The difference between this strategy and the average performance of our revised model is shown in Table 5.4.

Although our dynamic model outperforms the buy and hold strategy on average over the entire 45 years, the buy and hold strategy is significantly better on the second and third time period. Therefore, it is concluded that our model does not give an improvement over a simple buy and hold strategy.

Time	Hit rate	Accumulated return	Sharpe ratio
1	0.063*	470.67*	0.178*
2	-0.007*	-11.86*	-0.003*
3	-0.027*	-159.18*	-0.034*
Mean	0.010^{Δ}	99.88^{Δ}	0.047^{Δ}

Table 5.4: Difference between the mean performance of our model and the buy and hold strategy. A * indicates that the difference is significant ($p < 5$), differences that could not be tested on significance are market with a $^{\Delta}$.

Best performing strategy

Last, it is studied whether the model can outperform the best of the individual strategies it combines. Of course, it cannot outperform the best strategy in the second and third time period. It performs worse than the buy and hold strategy on those times, and that is one of the strategies it combines. However, in Table 5.5 it can be seen that the dynamic model also performs, on average, significantly worse than the best strategy on the first time period.

This table shows the difference between the performance of our model, and the performance of the best strategy on each time period. The best performing strategies and their performance are equal to those in experiment 1. It can be seen that our dynamic model cannot outperform the best strategy on any of the tested time periods.

Time	Hit rate	Accumulated return	Sharpe ratio
1	-0.001*	-3.74*	-0.001*
2	0.008*	-26.38*	-0.007*
3	-0.027*	-159.18*	-0.034*
Mean	$-0.007^{\triangle}$	$-63.10^{\triangle}$	$-0.014^{\triangle}$

Table 5.5: Difference between the mean performance of our model and the best of the strategies it combines. A * indicates that the difference is significant ($p < 5$), differences that could not be tested on significance are market with a $^{\triangle}$.

5.3 Discussion

In this experiment, it was tested whether a multi-agent model that uses a modified ReDVaLeR algorithm to update the proportions of strategies in a population, can be used to predict future trends in stock index prices. It is shown that the model cannot outperform the best of the strategies it combines. Therefore, it does not make an improvement over using only one single good strategy.

Nevertheless, the model can outperform a random strategy on two of the three tested periods, and the buy and hold strategy on one time period. However, for an investment model to be useful it is important to have a stable performance over time. That is definitely lacking in the current model.

One factor contribution to this lack of performance is the fact that the proportions in the model do update, in some situations, very slow. The revision of the original model was made to make the proportions update fast when needed, and still slow when stability is needed. The speed with which the proportions change is indeed dynamic over time. The method used to decide whether strategies learn slow or fast seems to cause some problems, however.

Strategies whose real performance corresponds to the expected performance based on its proportion have an individual learning rate of less than one. This was chosen to make the proportions stable when the proportion represent the success of the strategy well. For example, during the period from 1960 to 1975, one strategy outperforms all others. It is desirable that this strategy consistently makes up a large part of the population even when it performs bad for some short time. Therefore, a small learning rate is desirable.

However, the way in which the difference between the expected and measure performance is calculated may cause a problem. When a strategy has both the largest proportion and the largest (long term) return, it learns slow. This is desirable when the proportion is large, it will prevent the proportion from dropping too much when its short term performance stays behind. Furthermore, in this way, the proportion does not grow too fast such that the gap between other strategies becomes too large to bridge when the financial market changes.

A problem occurs, however, when the proportion of the strategy is indeed the largest of the population, but is not large yet. This can, for example, occur at the very start of the simulation. The best strategy grows fast in the beginning, due to its high returns. After a short period of time, however, it becomes the largest strategy. Even though its proportion is still small, for example 15 or 20 percent. This can occur when all proportions are relatively close together. In that case both the proportion and the return of the strategy will be the largest, and therefore the proportion will only grow very slow and has difficulties growing large.

The same occurs, of course, at the bottom. Strategies that are already the smallest, but not yet very small, could have difficulty becoming even smaller, even when they are losing money. This causes proportions to fall slowly after a first short period where they become smaller than the rest.

This problem could possibly be solved by using another method to compute the individual learning rates than using ranks. For example, the size of the difference between the expected and measured performance could be used. However, this would require another method to relate the proportion and measured performance. Future research should study whether this is an effective solution to our problem.

All in all, our revised multi-agent model does not seem suitable to combine strategies that predict trends in the prices of stock indexes. This may be due to some imperfections of the method to evolve the proportions in the population. However, whether an improvement of this method to evolve the proportions would indeed increase the effectiveness of the model in predicting stock index trends should be studied in a future study.

6. Discussion

In this research, it is studied whether a multi-agent model in which the proportions of each type of agent are dynamically updated over time, using (a modified version of) the replicator equation, can be used to combine strategies that predict the trend of stock index prices. We specified three requirements such a model should fulfil in order to be useful as a combination model. The model should perform better than all the strategies it combines, better than a simple model with static proportions, and its performance should be robust.

It was expected that the model would be able to fulfil those requirements. This expectation was based on previous studies. First, in earlier research it is shown that the (time-variant) combination of strategies can outperform individual strategies (Huang & Lee, 2010). Second, multi-agent models that combine multiple types of agents can be used to predict stock prices (Gupta, Johnson, & Hauser, 2005). Last, the replicator dynamic has been shown to be a useful population dynamic to describe the evolution of prices on the stock market (Galla & Zhang, 2009) and can be used to make real-world predictions as well (Bolluyt & Comaniciu, 2018). The combination of all those facts led to our expectation. Nevertheless, the expectation that our model would be able to make good trend predictions did not come true.

In the first experiment, it was studied whether a multi-agent model using the replicator dynamic as population dynamic fulfilled the three requirements. Although it could outperform a random strategy and a buy and hold strategy, it could not outperform the best of the strategies it combined. Therefore, requirement one was not fulfilled and the study was ended.

In the second experiment, the same model was tested, but with an adaptation of ReDVaLeR as population dynamic instead of the replicator dynamic. This model was not able to improve upon the performance of the best strategy it combined as well. Therefore, we have not found an indication that a dynamic multi-agent model can be used to predict stock index trends.

These results contradict the expectations. There are multiple possible explanations for these results. A first possible explanation is, as said before, the slow updating of the proportions over time. This causes the model to lag behind the stock market quite a bit. It would be interesting for future research to see whether this problem can be solved. For example, by using an adaptive learning rate, or using a different method to calculate the individual learning rates.

Furthermore, some other characteristics of the model, unrelated to the evolution of the population, may have influenced the performance as well. One contributing factor could be the fact that the model can only buy or sell index funds. Although the individual strategies in our model can invest in a risk free obligation, the model itself cannot do this (except in the highly unlikely situation that exactly as many agents buy and sell in the simulation). In the model simulations, however, the risk free investment strategy makes up the largest part of the population during some time periods (see, for example, Figure 4.3A). This suggests that risk free investing is a good trading strategy during that time. However, as our model cannot make risk free investments it cannot take advantage of this information.

Another factor influencing the performance can be the simplicity of the used strategies. Some of those strategies performed quite bad during one or multiple time periods. The proportions of those bad strategies drop quite fast toward zero percent. This is similar to "trimming", the removal of the worst strategies from the combination. Trimming is shown to increase the performance of combination models (Timmermann, 2006). Therefore, the presence of some bad strategies is not necessary problematic. However, during the last time period, almost no strategy performs well. When there is no good strategy in the population (only less bad strategies) inevitably the growing strategies are relatively bad as well. This can have contributed to the lack of performance of the model.

In future research, it could be tested whether a dynamic multi-agent model using the replicator dynamic can be more effective when the above limitations are handled better. However, for now, we have to conclude that there is no evidence that such a model gives a good combination of strategies. This brings us a step closer to a complete view of the possibilities of multi-agent models, especially dynamic multi-agent models, for predicting stock prices. It was already known that some static multi-agent models can make good prediction. Our study showed that dynamic multi-agent models can in most situations perform better than random and a simple buy and hold strategy.

However, our study showed as well that a dynamic multi-agent model based on the replicator dynamic is far from perfect. It cannot outperform the best of the individual strategies it combines. Therefore, future studies to the possibilities of dynamic multi-agent models are necessary to get a better understanding of their performance. Can, for example, other population dynamics be used? Does such a model using another population dynamic outperform the individual strategies it combines?

The results of this study also make it clear that a minimal multi-agent model may not be enough. A more realistic representations of agents may be needed. As mentioned in the introduction, the field of behavioural finance is growing. There becomes more and more evidence that investors do not behave rational after all. They make use of lots of biases and heuristics, like overconfidence and anchoring (Rekik, Hachicha, & Boujelbene, 2014). Integrating those human fallacies into a multi-agent model can make this model a much better representation of reality. Which in turn can lead to a better performance in explaining, or even predicting, price movements on the financial market.

There are already some artificial market models that make use of the insights gained through behavioural finance (e.g. Rekik et al., 2014) and even some cautious studies that try to predict real market prices with such behavioural models (e.g. Neri, 2012). However, there is much left to discover about the possibilities of multi-agent models in understanding and predicting the formation of asset prices.

To conclude, more research is needed to discover the possibilities of behavioural finance in general, and multi-agent models in particular, for forecasting asset prices. A good first step may be to follow up on our current research and study whether an improved dynamic multi-agent model can indeed be useful as a combination model of investment strategies. For now, it should be concluded that there is no indication that multi-agent models using (a modified version of) the replicator equation to update the agent's proportions over time are useful in combining investment strategies.

References

Antonello, R., & Silveira, R. A. (2010). Multiagent systems in stock index prediction. In *Trends in practical applications of agents and multiagent systems* (pp. 563–571). Springer.

Banerjee, B., & Peng, J. (2004). Performance bounded reinforcement learning in strategic interactions. In *Aaai* (Vol. 4, pp. 2–7).

Beyaz, E., Tekiner, F., Zeng, X.-j., & Keane, J. (2018). Comparing technical and fundamental indicators in stock price forecasting. In *2018 ieee 20th international conference on high performance computing and communications; ieee 16th international conference on smart city; ieee 4th international conference on data science and systems (hpcc/smartcity/dss)* (pp. 1607–1613).

Bolluyt, E. D., & Comaniciu, C. (2018). Dynamic influence on replicator evolution for the propagation of competing technologies. *IEEE Transactions on Evolutionary Computation*.

Challet, D., & Zhang, Y.-C. (1998). On the minority game: Analytical and numerical studies. *Physica A: Statistical Mechanics and its applications*, *256*(3-4), 514–532.

Cressman, R., & Tao, Y. (2014). The replicator equation and other game dynamics. *Proceedings of the National Academy of Sciences*, *111*(Supplement 3), 10810–10817.

Dea, S., Gondhib, N. R., Manglac, V., & Pochirajud, B. (2010). Success/failure of past trades and trading behavior of investors.

Edmonds, B., & Moss, S. (2001). The importance of representing cognitive processes in multi-agent models. In *International conference on artificial neural networks* (pp. 759–766).

Elliott, G., & Timmermann, A. (2005). Optimal forecast combination under regime switching. *International Economic Review*, *46*(4), 1081–1102.

Ellis, C. A., & Parbery, S. A. (2005). Is smarter better? a comparison of adaptive, and simple moving average trading strategies. *Research in International Business and Finance*, *19*(3), 399–411.

Farmer, J. D., & Joshi, S. (2002). The price dynamics of common trading strategies. *Journal of Economic Behavior & Organization*, *49*(2), 149–171.

Galla, T., & Zhang, Y.-C. (2009). Minority games, evolving capitals and replicator dynamics. *Journal of Statistical Mechanics: Theory and Experiment*,

2009(11), P11012.
Gou, C. (2005). Predictability of shanghai stock market by agent-based mix-game model. In *2005 international conference on neural networks and brain* (Vol. 3, pp. 1651–1655).
Grothmann, R. (2003). *Multi-agent market modeling based on neural networks* (Unpublished doctoral dissertation). University of Bremen, Germany.
Gupta, N., Hauser, R., & Johnson, N. (2005). Deducing the multi-trader population driving a financial market. *SPIE Complex Systems*, *6039*. doi: 10.1117/12.638334
Gupta, N., Hauser, R., & Johnson, N. F. (2007). Inferring the composition of a trader population in a financial market. In *Econophysics of markets and business networks* (pp. 99–113). Springer.
Gupta, N., Johnson, N., & Hauser, R. (2005). *Forecasting financial time-series using artificial market models* (Tech. Rep.).
Huang, H., & Lee, T.-H. (2010). To combine forecasts or to combine information? *Econometric Reviews*, *29*(5-6), 534–570.
Katahira, K., Chen, Y., Hashimoto, G., & Okuda, H. (2019). Development of an agent-based speculation game for higher reproducibility of financial stylized facts. *arXiv preprint arXiv:1902.02040*.
Keim, D. B., & Madhavan, A. (1995). Anatomy of the trading process empirical evidence on the behavior of institutional traders. *Journal of Financial Economics*, *37*(3), 371–398.
LeBaron, B., et al. (2001). A builder's guide to agent-based financial markets. *Quantitative Finance*, *1*(2), 254–261.
Lee, T.-H. (2011). Combining forecasts with many predictors. *Advances in economic forecasting. Kalamazoo, MI: WE Upjohn Institute for Employment Research*, 149–172.
Liu, G., & Wang, X. (2019). A new metric for individual stock trend prediction. *Engineering Applications of Artificial Intelligence*, *82*, 1–12.
Maharaj, E. A., D'Urso, P., & Caiado, J. (2019). *Time series clustering and classification.* CRC Press.
Malkiel, B. G., & Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, *25*(2), 383–417.
Marsili, M. (2001). Market mechanism and expectations in minority and majority games. *Physica A: Statistical Mechanics and its Applications*, *299*(1-2), 93–103.
Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2015). *Introduction to time series analysis and forecasting.* John Wiley & Sons.
Neri, F. (2012). Learning predictive models for financial time series by using agent based simulations. In *Transactions on computational collective intelligence vi* (pp. 202–221). Springer.
Petrusheva, N., & Jordanoski, I. (2016). Comparative analysis between the fundamental and technical analysis of stocks. *Journal of Process Management. New Technologies*, *4*(2), 26–31.
Rekik, Y. M., Hachicha, W., & Boujelbene, Y. (2014). Agent-based modeling and investors' behavior explanation of asset price dynamics on artificial

financial markets. *Procedia Economics and Finance*, *13*, 30–46.
Sharpe, W. F. (1994). The sharpe ratio. *Journal of portfolio management*, *21*(1), 49–58.
Sorropago, C., et al. (2014). Behavioral finance and agent based model: the new evolving discipline of quantitative behavioral finance?
Stasinakis, C., & Sermpinis, G. (2014). Financial forecasting and trading strategies: a survey. In *Computational intelligence techniques for trading and investment* (pp. 42–60). Routledge.
Thimmaraya, R., & Masuna, V. (2012). A novel quantitative behavioral framework for financial markets prediction. In *61st midwest finance association conference, february.*
Timmermann, A. (2006). Forecast combinations. *Handbook of economic forecasting*, *1*, 135–196.
Wieringa, M. A. (2001). *Introduction to adaptive systems.*
Yoshihara, A., Fujikawa, K., Seki, K., & Uehara, K. (2014). Predicting stock market trends by recurrent deep neural networks. In *Pacific rim international conference on artificial intelligence* (pp. 759–769).