In this article we describe the implementation of a diversified investment strategy using 25 intelligent agents. Each agent utilizes several data mining models and other artificial intelligence techniques to autonomously day trade an American stock. The agents were individually tested with out-of-sample data corresponding to the period between February of 2006 and June of 2010, and most achieved an acceptable performance. By integrating the 25 agents in a multi-agent system, we were able to obtain much better results (according to the return and maximum drawdown metrics); this leads us to believe that it might be possible to use one such system in the creation of a profitable hedge fund in which the investment decisions can be made without human intervention.

Keywords — financial trading; autonomy; intelligent agents

I. INTRODUCTION

The idea that it is possible to obtain above average returns by speculating in financial markets is far from consensual. The biggest claim against speculative trading was made by Fama [1] in his famous efficient market hypothesis, which postulates that a financial instrument’s price always fully reflects all the information available. This hypothesis implies that neither the instrument’s most up-to-date financial details nor its historical prices can be utilized to forecast its future price. Hence, if prices cannot be predicted, a trader’s success will always be a question of luck rather than skill. While this hypothesis cannot be completely disregarded, there are currently many studies that show empirical evidence against it. In particular, there are plenty of articles describing the successful use of data mining models in the prediction of financial data. Several different approaches have been tried, from stock price forecasting based on the categorization of press releases using support vector machines [2], to predicting exchange rates using artificial neural networks [3] or hybrid systems [4]. The mechanisms described in these studies are mostly intended for the development of simple tools to help human traders make trading decisions. Articles describing the development of systems intended to actually replace human traders altogether are not as abundant. Some simple agents created for this purpose have been presented in the Penn-Lehman Automated Trading Project [5]. However, even if these agents can open and close trades on their own, they cannot be considered truly autonomous, due to their inability to learn or adapt to new market conditions. Without these skills, they are unlikely to be profitable in the long run.

II. THE TRADING AGENT ARCHITECTURE

In order to supersede its human counterpart, an intelligent trading agent will need to be completely autonomous, which as we see it implies complying with the following requirements:

- it should be able to decide when to buy, short sell or close open trades,
- it should be able to perform money and risk management,
- it should be able to keep learning over time, even as it trades,
- it should be able to adapt to changes in market conditions; in particular, it should be “smart” enough to stop trading when the market becomes less predictable, and to restart trading when conditions improve.

Our research was motivated by this current lack of well-founded studies on the subject of true autonomous trading in public literature. Our objective will be to create intelligent trading agents that can be used to devise an automatic investment system to emulate a real life autonomous hedge fund, in which human intervention can be relegated to simple managerial tasks.

Figure 1. The trading agent architecture.
The agent architecture is composed of three modules:
- the prediction module, implemented using an ensemble of data mining models;
- the empirical knowledge module, implemented using a case-based reasoning system;
- the domain knowledge module, implemented using an expert system.

The prediction module is responsible for deciding if a stock should be bought or short sold, based on the predictions of the data mining models in the ensemble. Each model tries to predict if the stock price will increase or decrease throughout the next trading day (from open to close). The models’ predictions are aggregated into a single forecast, which the agent uses to pick the direction of the trade: if the forecast dictates that the stock price will increase, then the module’s decision is to buy the stock when the market opens, and sell it when it closes; otherwise, if a price decrease is forecasted, the module’s decision is to short sell the stock at the open and cover at the close. The algorithm behind the decisions of the prediction module is shown in figure 2. Notice that the weight of the vote of each model in the module’s decision is given by either its long profit factor (if it predicts a price increase) or its short profit factor (if it predicts a price decrease); both values are calculated before each forecast using equations 2 and 3.

While the prediction module allows the agent to automatically decide when to buy or short sell a financial instrument, that in itself is not sufficient to make it completely autonomous. Besides being able to decide the direction of each trade (i.e., if it should go long or short), the agent also needs to be capable of deciding how much to invest (i.e., the size of each trade), so that it can decrease the investment amount or even stop trading when the perceived risk is higher. That is the purpose of the empirical knowledge module. For each potential trade, it can set the investment amount to three different sizes: if the trade is expected to be profitable, a standard, user-defined trade size is used; if there are doubts regarding the profitability of the trade, half the standard trade size is used; finally, if the trade is expected to be unprofitable, its size is set to zero, which means the agent will not place that trade. The empirical knowledge module gets its name from the fact that it uses information from previous trades to decide on the amount to invest in new trades. The main component of the empirical knowledge module is a case-based reasoning system. In this system, each case corresponds to a trade that was previously executed by the agent, and contains the following information:
- the direction predicted by the prediction module,
- the direction predicted by each model in the prediction module’s ensemble,
- the return of the trade.

This mechanism tries to capitalize on the bigger profitability associated with certain combinations of the models’ predictions. For example, our empirical results show that trades carried out when all the models make the same prediction, i.e., all predict a price increase or all predict a price decrease, are consistently more profitable than those

\[
\text{Overall PF} = \frac{\sum \text{returns of profitable trades}}{\sum \text{returns of unprofitable trades}} - 1 \quad (1)
\]

\[
\text{Long PF} = \frac{\sum \text{returns of profitable long trades}}{\sum \text{returns of unprofitable long trades}} - 1 \quad (2)
\]

\[
\text{Short PF} = \frac{\sum \text{returns of profitable short trades}}{\sum \text{returns of unprofitable short trades}} - 1 \quad (3)
\]

performed when the predictions are mixed. The algorithm governing the empirical knowledge module’s decisions is shown in figure 3.

Both the prediction and the empirical knowledge modules were devised in way that allows the agents to keep learning while they trade. However, there will always be some expert knowledge that the agents will not be able to pick up from practice. The domain knowledge module was
created to overcome this problem. As its name implies, its main responsibility is to use domain-specific knowledge to make trading decisions. This module consists of a rule-based expert system in which trading experts can insert rules to guide the trading activity of the agents. These rules can be related to many different aspects of trading; for example, they can define low liquidity periods in which the agents should not trade, or they can be used to make the agents close open trades when a certain profit or loss is reached. In the agent architecture that we propose, the domain knowledge module is responsible for making the final trading decisions, by taking into account the prediction module’s recommendations to buy or short sell the financial instrument, the empirical knowledge module’s trade size suggestions, and its own expert rules.

III. THE MULTI-AGENT HEDGE FUND

We used the architecture described in the previous section to implement 25 intelligent agents, each being responsible for trading a different American stock. All the agents used the exact same settings, with the exception of the models in their ensembles. In order to prevent overfitting and biased results, we created a program to automatically select an ensemble of 11 data mining models for each agent. This software trained hundreds of models using the available price data up to October of 2005, and picked the 11 best, based on the prediction heterogeneity of the group, and on the models’ individual performances when tested with data for the period going from November of 2005 till December of 2005.

After creating the 25 agents, we tested their trading skills using out-of-sample data for the period between February of 2006 and June of 2010. Their accumulated return during this period is shown in figures 4 through 8. Some of them achieved an excellent performance, while others did not do so well. This is more or less what we would expect to see in

---

**Figure 3.** Pseudo-code for the empirical knowledge module algorithm.

GET THE ENSEMBLE’S AND THE MODELS’ PRICE DIRECTION FORECASTS FOR THE TARGET PERIOD FROM THE PREDICTION MODULE.

RETRIEVE FROM THE DATABASE ALL THE CASES WITH THE SAME COMBINATION OF PREDICTIONS.

WHILE THE NUMBER OF RETRIEVED CASES IS LOWER THAN A USER-DEFINED MINIMUM:

REMOVE THE LAST MODEL’S PREDICTION FROM THE SEARCH AND RETRIEVE THE CASES AGAIN.

CALCULATE THE OVERALL PROFIT FACTOR OF THE RETRIEVED CASES (EQUATION 1).

IF THE PROFIT FACTOR IS GREATER THAN OR EQUAL TO A USER-DEFINED THRESHOLD:

MAKE THE TRADE SIZE EQUAL TO THE STANDARD SIZE.

ELSE IF IT IS LOWER THAN ANOTHER USER-DEFINED THRESHOLD:

MAKE THE TRADE SIZE EQUAL TO ZERO.

ELSE:

MAKE THE TRADE SIZE EQUAL TO HALF THE STANDARD SIZE.

ONCE THE TRADE IS CLOSED, INSERT A NEW CASE IN THE DATABASE.

---

**Figure 4.** Accumulated returns of the agents that traded the stocks AA, APL, ADBE, BAC and CAL (trading costs included).

**Figure 5.** Accumulated returns of the agents that traded the stocks CSCO, DELL, DIS, GE and GOOG (trading costs included).

**Figure 6.** Accumulated returns of the agents that traded the stocks HD, IBM, INTC, JNJ and KFT (trading costs included).

**Figure 7.** Accumulated returns of the agents that traded the stocks KO, MCD, MRK, MSFT and NVDA (trading costs included).
a traditional investment company: some traders perform better than others, but ultimately what really matters is the company’s return as a whole. In order to emulate the inner-workings of this type of company, we integrated the 25 intelligent agents in a multi-agent system, and made them share the trading resources. The system’s main advantage is that it eliminates much of the trading risk associated with each individual agent, because the losses of the worst agents are compensated with the gains of the best. Compared to a traditional investment firm, our small AI-based hedge fund prototype has the advantage of requiring almost no human intervention. The accumulated return obtained by this multi-agent system during the out-of-sample period is shown in figure 9.

Figure 8. Accumulated returns of the agents that traded the stocks PFE, T, VZ, WMT and XOM (trading costs included).

Figure 9. Accumulated return of the multi-agent trading system (before and after the trading cost are subtracted), compared with the buy-and-hold strategy.

Our trading results show that the multi-agent system obtained a return of 29.9% in four and a half years of simulated trading, with a maximum drawdown of 4.4%. While acceptable, this is far from an extraordinary performance. Nevertheless, there are two reasons why these results can be considered very promising. First of all, the system was able to trade profitably in 2008, an eventful year in which several major financial companies collapsed due to the subprime mortgage crisis. The impact of this crisis in the stock market is very clear in the results of the simpler buy-and-hold strategy: it returned a profit of 18.6%, but with a dramatic maximum drawdown of 51.7%. The fact that the multi-agent system was able to maintain a low maximum drawdown even in these challenging conditions is, therefore, a noteworthy accomplishment. The other reason why its results can be considered promising is that they do not account for the use of leverage or compounding. Since the system’s strategy seems to be relatively safe (given its low maximum drawdown even when the market makes extreme moves), leverage and compounding should allow it to obtain a much better return, without incurring too much risk. Thus, the proposed agent architecture and multi-agent system show a lot of potential, and should be of practical interest to the investment industry.

IV. CONCLUSION

In this article we described the implementation of a diversified investment strategy using a multi-agent system. This strategy showed promising results in simulated trading, leading us to believe that the proposed system could prove useful for autonomous real life trading. Nevertheless, there are still several improvements that could be made to the trading agents. For example, in our experiments we only utilized price and time-based attributes to train the agents’ data mining models, as well as some simple technical analysis indicators, such as the RSI, the Williams %R and the ROC. It is possible that using better attributes might improve the models’ accuracy, which should in turn make the agents more profitable. Also, the proposed multi-agent system could be further improved by increasing the number of agents. This would increase the investment diversification (hence decreasing the trading risk), especially if the new agents were prepared to trade different financial instruments, such as futures or currency pairs.

While the test results obtained by the multi-agent system do not guarantee that it will be successful in the future (because past performance is not a guarantee of future returns), we believe its performance was at least interesting enough to warrant more research on this topic. Our results indicate that, with a sufficient number of agents, it might be possible to create a reasonably profitable system to operate as a real life hedge fund, based solely on data mining and artificial intelligence, and requiring virtually no human intervention in the trading room.

REFERENCES