Journal of Economic Dynamics & Control 34 (2010) 1153–1170

Contents lists available at ScienceDirect

Journal of Economic Dynamics & Control

journal homepage: www.elsevier.com/locate/jedc

JOURNAL OF Economic Dynamics & Control

Heterogeneous trading strategies with adaptive fuzzy Actor–Critic reinforcement learning: A behavioral approach

Stelios D. Bekiros [*]

European University Institute, Via Delle Fontanelle 10, 50014 San Domenico di Fiesole, FI, Italy

ARTICLE INFO

Article history:
Received 8 February 2009
Accepted 20 January 2010
Available online 6 February 2010

JEL classification:
G17
C45
C63

Keywords:
Agent-based modeling
Technical trading
Reinforcement learning
Fuzzy inference
Bounded rationality

ABSTRACT

The present study addresses the learning mechanism of boundedly rational agents in the dynamic and noisy environment of financial markets. The main objective is the development of a system that “decodes” the knowledge-acquisition strategy and the decision-making process of technical analysts called “chartists”. It advances the literature on heterogeneous learning in speculative markets by introducing a trading system wherein market environment and agent beliefs are represented by fuzzy inference rules. The resulting functionality leads to the derivation of the parameters of the fuzzy rules by means of adaptive training. In technical terms, it expands the literature that has utilized Actor–Critic reinforcement learning and fuzzy systems in agent-based applications, by presenting an adaptive fuzzy reinforcement learning approach that provides with accurate and prompt identification of market turning points and thus higher predictability. The purpose of this paper is to illustrate this concretely through a comparative investigation against other well-established models. The results indicate that with the inclusion of transaction costs, the profitability of the novel system in case of NASDAQ Composite, FTSE100 and NIKKEI255 indices is consistently superior to that of a Recurrent Neural Network, a Markov-switching model and a Buy and Hold strategy. Overall, the proposed system via the reinforcement learning mechanism, the fuzzy rule-based state space modeling and the adaptive action selection policy, leads to superior predictions upon the direction-of-change of the market.

1. Introduction

Heterogeneous agents approach challenges the conventional representative, rational agent framework. Heterogeneity in expectations can lead to market instability and complicated dynamics of prices, which are driven by endogenous market forces. Simon (1957) argued that boundedly rational agents using simple rules-of-thumb for their decisions under uncertainty, provides a more accurate and realistic description of human behavior than perfect rationality with optimal decision rules. Ever since the introduction of the Efficient Markets Hypothesis, fully rational agents were considered the driving forces of markets, which in turn operated in a way to aggregate and process the beliefs and demands of traders reflecting all available information (Fama, 1970, 1991). But the empirical evidence from financial markets was not in full accordance with the Efficient Markets Hypothesis. The alternative behavioral model was based on relaxing strict rational agent assumptions and introducing market frictions. The key arguments of behavioral agent-based models reported by

[*] Tel.: +39 55 4685 698; fax: +39 55 4685 804.
E-mail address: Stelios.Bekiros@eui.eu

0165-1889/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.jedc.2010.01.015

Hommes (2001, 2006) are closely related to Keynes view that "expectations matter", to Simon's result that humans are boundedly rational and to the Kahneman–Tversky analysis in psychology that individual behavior under uncertainty can be described by simple heuristics and biases.

In view of empirical studies that stock prices can be predicted with a fair degree of reliability advocates of Efficient Markets Hypothesis (e.g., Fama and French, 1995) claim that such results are based on time-varying-equilibrium expected returns generated by rational pricing in an efficient market which compensates for the level of risk undertaken. On the contrary, opponents (e.g., La Porta et al., 1997; Shiller, 2002) argue that predictability reflects the psychological factors and fashions or "fads" of irrational investors in a speculative market. This irrational behavior has been emphasized by Shleifer and Summers (1990) and Black (1986) in their exposition of noise traders who act on the basis of imperfect information and consequently cause prices to deviate from their equilibrium values. Arbitrageurs dilute a minor part of these shifts in prices, yet the major component of deviation is tradable. Moreover, Black claimed that noise traders play a useful role in promoting market liquidity. Overall, there are two types of agents in heterogeneous markets: "fundamentalists", who base their expectations upon dividends, earnings, growth or even macroeconomic factors, and "chartists" (noise traders and technical analysts) who instead base their trading strategies upon historical patterns and heuristics and try to extrapolate trends in future asset prices (Brock and Hommes, 1998).

Currently, technical analysis is being widely used among market practitioners as an effective technique to earn significant profits from financial trading (Kirkpatrick and Dahlquist, 2007; Murphy, 1999; Kaufman, 1998). Recent work based on a behavioral finance perspective has revealed that the application of technical analysis in trading can consistently produce high returns, albeit it is still considered a "pseudoscience" compared with conventional fundamental analysis (Irwin and Park, 2007; Camillo, 2008; Plummer and Ridley, 2003; Lo et al., 2000). This is due to the inherent subjectivity and a general lack of established guidelines to systematically determine the amount of relevant historical information as well as the optimal parameters of the technical rules employed. Obviously, the entire setup is heuristically determined and heavily dependent on traders' experience and beliefs (Lawrence and O'Connor, 1992; Bolger and Harvey, 1995). In this framework, "reinforcement learning" methodology inspired by psychology and commonly used in artificial intelligence and heterogeneous agent modeling, appears to provide an ideal approach to model knowledge, experience, intuition and consequently the decision-making process under which technical analysts operate.

2. Machine learning and intelligent control

In heterogeneous markets the major challenge for chartists is the development of new models, or the modification of existing methods, that would enhance predictability particularly in an environment with dynamic time-varying patterns. Conventional modeling based on econometric methods (e.g., ARMA, ARCH-GARCH, VAR), spectral analysis (e.g., Fourier transforms, periodograms), stochastic processes (e.g., random walks), or control systems (e.g., linear time invariant models) does not always perform satisfactorily in financial applications. The reason is that financial data are often characterized by chaotic behavior, extreme events and nonlinear components.

Machine learning techniques have been employed in control systems in modeling nonlinearities and especially towards simulating human behavior. In general, machine learning refers to systems capable of autonomous acquisition and integration of knowledge. It is a very active area of research in Artificial Intelligence, concerned with developing learning algorithms in real-world complex systems. In many applications, taking for instance financial markets, the modeling difficulty is inbuilt because the state space is enormous, the behavior of agents complex and conflicting and the system environment cannot be described by a linear analytical model as in classic model-based control. Hence it is crucial and essential for the agents to have the ability to learn and adapt. Conventional control has been implemented in the past by human expert operators. Many times it proved to be time-consuming, divergent, erroneous and unfit to real-time demands. As a solution the implementation of intelligent control combines nonlinear, adaptive and stochastic methods to enhance system autonomy, reliability and efficiency (Antsaklis and Passino, 1993). Powerful techniques from intelligent control include evolutionary algorithms, neural networks, fuzzy logic, reinforcement learning as well as hybrid approaches.

Neural networks have been extensively used in learning literature, as well as in nonlinear time series modeling and in function approximation. They are parallel computational models comprising input and output vectors as well as processing units (neurons) interconnected by adaptive connection strengths (weights), trained to store the "knowledge" of the network. Bertsekas and Tsitsiklis (1996) and Barto et al. (1983) report the use of neural networks in learning control problems. Adya and Collopy (1998) demonstrated the advanced predictive ability of neural networks for time series forecasting, while White (1989) and Kuan and White (1994) suggested that traditional time series techniques for forecasting have reached their limitation in applications with nonlinearities within the data sets. The function approximation properties have been thoroughly investigated by many authors. The results in Cybenko (1989), Funahashi (1989), Hornik (1991), Hornik et al. (1989, 1990), Gallant and White (1988, 1992) and Hecht-Nielsen (1989) demonstrated that feedforward networks with sufficiently many hidden units and properly adjusted parameters can approximate any function to any desired degree of accuracy. Poddig (1993) applied a feedforward neural network to predict the exchange rates between American Dollar and Deutsche Mark, and compared results to regression analysis. Other examples using neural networks in financial markets include Gençay (1998b), Green and Pearson (1994), Manger (1994), Rawani et al. (1993), Weigend (1991), Yao et al. (1996) and Zhang (1994). However, in financial applications neural networks utilize in input space only quantitative factors, such as stock returns, indices and other economic measures. A number of qualitative factors, such as macroeconomic or political effects as well as traders' psychology, may seriously influence the market trend. Thus, it is important to capture this "unstructured" knowledge.

Fuzzy logic has been implemented initially in the area of control systems and only recently in economic applications with highly promising results. It provides a means of representing uncertainty with imprecise data. In that sense it can be an excellent tool for agent-based learning and decision-making under uncertainty. Specifically, in fuzzy systems numeric variables (inputs and outputs) are translated into linguistic terms. *Fuzzy inference rules* represented by IF-THEN statements are specified to associate fuzzy input to fuzzy output. For instance, these rules could comprise an efficient mechanism of incorporating heterogeneous agent beliefs or unstructured knowledge in the form of rules-of-thumb. Fuzzy logic has been applied to classification, process simulation and decision support systems, as an effective means of modeling human experience and intuition (Sugeno, 1988; Kosko, 1992; Klir and Yuan, 1995; Jamshidi et al., 1997; Al-Shammari and Shaout, 1998). Financial applications have also been reported (Altrock, 1997; Tay and Linn, 2001; Gradojevic, 2007). One important advantage of fuzzy inference systems is their linguistic interpretability. When implementing fuzzy systems, the focus is paid on modeling fuzziness and linguistic vagueness using membership functions. Nevertheless, while the fuzzy logic-based approach shows promising results the process to construct a fuzzy system is basically subjective and depends on some ad-hoc assumptions. Some techniques from neural networks or reinforcement learning can be employed in order to "calibrate" fuzzy rules by means of adaptive training. Further details on reinforcement learning methodology are provided in the next section.

In summary, this study addresses the learning issue for boundedly rational agents in the highly dynamic and noisy environment of financial markets. The main objective is the development of a learning algorithm that decodes the knowledge acquisition mechanism and decision-making process of technical analysts. In this direction an approach combining reinforcement learning and fuzzy inference is followed. The remainder of this paper is organized as follows: Section 3 provides a brief overview of reinforcement learning. Moreover, based on the most recent developments in the field, fuzzy reinforcement learning is further described. In Section 4, a new adaptive fuzzy Actor–Critic reinforcement learning system is introduced. Then, in Section 5 other competing models used in this study are described and the empirical results are presented in Section 6. Finally, Section 7 provides concluding remarks.

3. Fuzzy reinforcement learning

3.1. An introduction to reinforcement learning

Reinforcement learning is a model-free computational technique aiming at the automation of goal-directed decision making in a dynamic environment. It is a sub-field of machine learning and it has been successfully applied to complex problem-solving (Kaelbling et al., 1996). Sutton and Barto (1998) provide a descriptive definition: "*Reinforcement learning is learning what to do, how to map situations to actions, so as to maximize a numerical reward signal.*" On the contrary to well-known conventional learning algorithms concerned with how to model an input/output mapping, the reinforcement learning approach tries to reveal emerging behaviors without other information but a scalar signal, i.e., the reinforcement. The "agent"—a general term that designates a system under training—uses this signal to determine an optimal policy which results in maximum total reward. Under this kind of training, the agent is continuously analyzing the consequences of its actions through trial-and-error interaction with the environment and learning occurs after a sequence of "episodes" (Kaelbling et al., 1996). The reinforcement learning approach comprises the agent, the environment, the state information, the actions and the reward signal. Specifically, at each learning step t the agent perceives the current state s_t of the environment and the corresponding reward r_t. Based on the state information and the reinforcement signal the agent updates the "quality" of its actions and selects an action α_t as a response to the environment. The transition to a new state s_{t+1} is determined by the action α_t and this event-cycle continues until the end of one learning episode. Additionally, the dynamics of the reinforcement learning system are determined based on three elements, namely the selection policy, the value function and the reward function (Sutton and Barto, 1998).

Formally, the basic reinforcement learning model consists of a set of environment states **S**, a set of agent actions **A** and a scalar reward/reinforcement function R. The environment is typically formulated as a finite-state Markov decision process (MDP), thus reinforcement learning algorithms are highly related to dynamic programming techniques. State transition probabilities and reward probabilities in the MDP are typically stochastic but stationary over the course of the problem. The transition of the environment from one state to the other is probabilistic in nature and p_{ij} is the transition probability from state i to state j. If $\alpha \in \mathbf{A}$ then $p_{ij}(\alpha) = Pr(s_{t+1} = s_j | s_t = s_i, \alpha(t) = \alpha)$. In every state an agent chooses an action according to a certain policy, thus $\alpha_t = \pi(s_t)$. Each policy has a value function V^π which represents the "quality" of the agent behavior and in order to calculate V^π the reward function should be specified. Usually a discounted model such as $R_t = \sum_{k=0}^{\infty} \xi^k r_{t+k}$ with $0 \leq \xi \leq 1$ is used, or alternatively an error function is employed (Kaelbling et al., 1996). Based on the discounted model for instance, the value function V^π can be recursively written as $V^\pi(s) = R(s, \pi(s)) + \xi \sum_{s_{t+1} \in S} p_{ss_{t+1}}(\pi(s)) V^\pi(s_{t+1})$, where $R(s, \alpha) = E\{r(s, \alpha)\}$ is the expected reward when one applies an action and $p_{ss_{t+1}}$ is the probability to pass from state s to s_{t+1}.

Considering that an optimal policy π^* exists, then the Bellman optimality equation is satisfied

$$V^{\pi^*}=V^*(s)=\max_{a\in A_s}\left\{R(s,\alpha)+\xi\sum_{s_{t+1}}P_{ss_{t+1}}(\alpha)V^*(s_{t+1})\right\}\quad \forall s\in S$$

Two methods are used in order to calculate the optimal policy, namely *value iteration* and *policy iteration.* In value iteration, the action of optimal value for every state is determined via iteration. The value of an action is defined as $Q^{\pi}(s,\alpha)=R(s,\alpha)+\xi\sum_{s_{t+1}}p_{ss_{t+1}}(\alpha)V^{\pi}(s_{t+1})$ and the Bellman optimality becomes $V^*(s)=\max_{a\in A_s}Q^*(s,\alpha)$. Unlike value iteration algorithm, policy iteration deals directly with the policy itself. The value function is computed for an intermediate policy not for all possible policies (different choice of actions) in a state. Usually it needs a smaller number of iterations than value iteration technique.

A common categorization in reinforcement learning approaches is that of *model-free* (direct) and *model-based* (indirect), with the first being more generally applicable. Some approaches are based on value iteration such as Q-learning and SARSA (Watkins, 1989) and others on policy iteration such as Actor–Critic learning (Sutton and Barto, 1998). Additionally, temporal difference (*TD*) technique (Sutton, 1989) is used by all algorithms in model-free reinforcement learning. It deals with an approximation of value function V^*. As the action a_t passes from s_t to s_{t+1} with reinforcement r_t, the new evaluation becomes $r_t+\xi\hat{V}_t(s_{t+1})$. The basic $TD(0)$ update rule (takes only into account the new state) is given by $\hat{V}_{t+1}(s_t)=\hat{V}_t(s_t)+\eta\{r_t+\xi\hat{V}_t(s_{t+1})-\hat{V}_t(s_t)\}$, where η is the learning rate. The difference between the two successive evaluations, called *TD Error*, is $E_t=r_t+\xi\hat{V}_t(s_{t+1})-\hat{V}_t(s_t)$. If the learning rate decreases sufficiently slowly towards zero, the successive evaluations of the value converge to the optimal value function.

Q-learning is one of the most popular reinforcement learning methods developed by Watkins (1989) and is based on $TD(0)$. It involves finding state-action qualities rather than just state values. The estimation of optimal state-action value function is

$$\hat{Q}_{t+1}(s_t,\alpha_t)=\hat{Q}_t(s_t,\alpha_t)+\eta\cdot\{r_t+\xi\max_{\alpha}\hat{Q}_t(s_{t+1},\alpha_{t+1})-\hat{Q}_t(s_t,\alpha_t)\}$$

where $0<\eta\leq 1$ is the learning rate. Q-learning considers that the next action is chosen based on highest Q-value. Regarding on-line performance, Q-values will converge to optimal values irrespectively of the balance between exploration (of uncharted territory) and exploitation (of current knowledge) in action selection mechanism (Kaelbling et al., 1996).

Actor–Critic learning is another model-free method. The Critic represents an approximation of the value function of the current policy running in Actor component. Both Critic and Actor parts can learn simultaneously: the Actor tries to develop the optimal policy at time step t with respect to the Critic value, while at the same time Critic tries to learn the value function of the current policy initiated in Actor. Critic learning uses an update rule of parameters involved in action selection and usually is based on the *TD error*. The convergence is faster compared with Q-learning. However, many times the standard approach of approximating a value function and determining a policy from it has so far proven theoretically intractable. Actor–Critic approach also utilizes linear/nonlinear function approximation which is essential to reinforcement learning in order to deal with the curse of dimensionality both in discrete and continuous time. This modified version called *direct policy* or *policy gradient method*, introduced by Sutton et al. (2000), defines the policy as a parameterised differentiable function updated with respect to the policy parameters. The vector of policy parameters is updated approximately proportional to the gradient as follows:

$$\Delta\mathbf{z}_t\approx\eta(\partial r_t/\partial\mathbf{z}_t)\tag{1}$$

where $\mathbf{z}$ is the parameter vector, η learning rate and r_t the reinforcement signal, which is usually represented by the prediction error (e.g., MSE or RMSE, etc.), albeit not necessarily. Consequently, if $f_u(s,\alpha),\forall\alpha\in\mathbf{A},\forall s\in\mathbf{S}$ is an approximation function with parameters $\mathbf{u}$, the error gradient after partial derivation is given by $(\partial r_t/\partial\mathbf{z}_t)=(\partial r_t/\partial f_{u_t})(\partial f_{u_t}/\partial\mathbf{u}_t)(\partial\mathbf{u}_t/\partial\mathbf{z}_t)$. It is proven that this method provides convergence to global optimal policy for a wide variety of algorithms based on Actor–Critic architecture (Sutton et al., 2000; Kimura and Kobayashi, 1998). In this context, nonlinear approximation functions from computational intelligence could be employed, such as feedforward neural networks (Gullapalli, 1992) or fuzzy logic models (Berenji, 1991, 1996; Glorennec, 1993). In the present study a *policy gradient* Actor–Critic method with the utilization of fuzzy reinforcement learning will be introduced.

3.2. Fuzzy logic and reinforcement learning

Fuzzy logic is embedded in reinforcement learning in order to enhance learning and adaptation of agent-based systems. Several studies report the "synergetic" advantages of the combined methodology. Glorennec (1993) proposed a fuzzy version of Q-learning, where an "optimal" rule set is extracted among different possible rules. Specifically, a rule "quality" is defined for each rule of each agent and a set of agents compete to "drive" the system. Eventually, only one agent is selected and its rule is activated. A similar algorithm was reported by Berenji (1996). Glorennec and Jouffe (1996) presented an alternative approach by introducing quality factors for actions associated with each rule. The action selection mechanism is "greedy" and performed via an exploration/exploitation trade-off. At each episode "exploitation" corresponds to the selection of the best policy, while at the same time all possibilities are tried (explored) in order to ensure that the learned behavior is globally and not locally optimal. In a more recent work by Er and Deng (2004) a new model is introduced capable of generating new rules and adding more linguistic partitions over the universe of discourse based on two criteria namely Mahalanobis distance (*M*-distance) and *TD* error. However, the computational load was reported high and eventually beyond real-time requirements. While most of the work is on fuzzy Q-learning, Glorennec and Jouffe (1997) and Jouffe (1998) developed Actor–Critic learning algorithms with the incorporation of fuzzy logic. The main objective is to increase efficiency under real-time requirements in case of on-line learning. They concluded that the utilization of fuzzy inference in Actor–Critic reinforcement learning presents many advantages such as universal approximation, the possibility to treat continuous state and action spaces, the speedup and optimal convergence of the training process, and most significantly the encapsulation of *a priori* knowledge and the interpretation of the acquired knowledge after the training phase. Apart from these studies there has not been further work on this topic.

In a fuzzy reinforcement learning system the mechanism of incorporating the heterogeneous beliefs of agents under imprecise knowledge can be implemented by *fuzzy inference rules*. Fuzzy learning, represented by IF-THEN statements, provides a very realistic model of agent's decision-making process. Technically, the general fuzzy inference architecture consists of the input, the rule layer and the output layer. In the input layer all the input variables are translated into fuzzy linguistic terms, e.g., "low" and "high", whereas in the Boolean formalism inputs have a crisp numeric value. Each term is described by a membership function, which estimates the "*degree*" to which a variable belongs to a fuzzy set. The fuzzy learning rules consist of two parts, namely "IF" and "THEN" part. The "IF" part uses an "AND" operator proposed by Zimmerman and Thole (1978), which represents the minimum value among all the validity values. The output fuzzy layer uses fuzzy membership functions for output variables. Finally, in the defuzzification layer, the output is converted from fuzzy variables back into crisp values. The aforementioned structure utilizes the Mamdani (1977) approach of fuzzy learning. Alternatively, Sugeno's (1985) approach introduces linear dependences of each rule on the system's input variables, whereby no defuzzification process is required. The more general first-order Sugeno fuzzy model has rules of the form "IF x_1 is A AND x_2 is B THEN $z=h+cx_1+dx_2$", where (x_1,x_2) are the inputs, (A, B) are fuzzy sets and c, d and h are parameters. Because of the linear dependence of each rule on the system's input variables the Sugeno system is suited for modeling complex nonlinear systems by interpolating multiple linear models. The specific rule-based system has an antecedent part (parameters of the membership functions) and a corresponding consequent (polynomial parameters) for each rule. In this line of literature, the main technical issue is the choice of the linguistic terms for each fuzzy variable and the estimation of the consequent parameters, whereas the antecedent part is usually fixed and not "calibrated". In this way, the fuzzy inference rules depend on ad-hoc assumptions, are subjectively constructed and eventually the choice of membership functions depends on trial and error. Thereby, efficient training algorithms from neural networks can be utilized to "fine-tune" the fuzzy membership functions (Buckley and Hayashi, 1994; Lin and Lee, 1996; Nishina and Hagiwara, 1997).

In the next section a new "*adaptive*" fuzzy Actor–Critic reinforcement learning system is presented. The resulting functionality leads to the derivation of the parameters of fuzzy inference rules by means of adaptive training. This is implemented with novel modifications to match the Actor–Critic methodology. On the whole, the proposed setup advances the literature on heterogeneous learning in speculative markets by introducing a trading system based on Actor–Critic modeling, with the incorporation of beliefs and idiosyncratic behavioral patterns represented by fuzzy inference rules. In technical terms, it expands the literature that has utilized Actor–Critic reinforcement learning and fuzzy systems in agent-based applications, by presenting an adaptive fuzzy reinforcement learning approach that provides with accurate and prompt identification of market turning points and thus higher predictability. The purpose of this paper is to illustrate this concretely through a comparative investigation against other well-established nonlinear models. Moreover, beyond the existing practice that has commonly utilized first moment measures such as return lags, moving averages, etc. in the input and state space, it is demonstrated that the new model leads to enhanced forecastability via the incorporation of conditional volatility (second moment). The use of the standard model of price growth from financial theory in defining the fuzzy rules enables the state space to incorporate information from the expected rate of return as well as from the volatility diffusion parameter in order to optimally track the time-varying characteristics of the underlying variables.

4. A trading system based on adaptive fuzzy Actor–Critic learning

4.1. Preliminaries

The historical time-series comprise the environment for the reinforcement learning agent. The Actor–Critic learning strategy is then employed by the agent in order to derive the action policy based on the interaction with the environment. The mechanism is briefly described as follows: The reinforcement learning process is performed in an episodic fashion and each episode consists of a sequence of agent–environment interactions. An episode ends when the learning agent reaches a predefined terminal condition. The recent fluctuations of the underlying financial time-series are used to define the state for the reinforcement learning task. The time-varying price behavior is characterized by the expected return (first moment) and the conditional volatility (second moment), which implicitly "reflect" the prevailing trading conditions of the market. Following that, based on the current state, an optimal parameter vector representing the response (action) to the environment is selected by the Actor–Critic learning agent. The selected antecedent (Actor) and consequent (Critic) parameters are then used to define the technical trading rule to generate the policy (output of the fuzzy inference system)

Table 1
Elements of the adaptive fuzzy Actor–Critic reinforcement learning system.

Environment	The historical time-series accessed by the technical analyst
State	Eight distinct states corresponding to the fuzzy inference rules in order to characterize the underlying financial input variable based on expected return and conditional volatility
Policy	Output of the fuzzy inference system in the form of a parameterised nonlinear function updated with respect to action parameters, leading to a two-way trading decision (buy or sell)
Action	The selected optimal parameter values by the learning agent representing the response to the environment
Reward	The prediction accuracy measured by the forecasting mean squared error

and thus the two-way trading decision (buy or sell). The policy is represented by the nonlinear output function of the Sugeno fuzzy system, according to the *policy gradient approach* coined by Sutton et al. (2000). Subsequently, the quality of the calibrated parameter values (action) is evaluated by the learning agent via the prediction accuracy of the corresponding fuzzy model and specifically by computing the forecasting error, which is used as the reinforcement signal. In general, a small error indicates a good choice of the selected parameters and vice versa.

The inputs $\mathbf{x}$ of the adaptive fuzzy Actor–Critic system correspond to the returns of the previous days and the volatility daily changes, while the output y is the forecasted 1-day-ahead return. A specification with two lags of the logarithmic returns and one lag of the conditional volatility changes is selected.[1] In the proposed architecture two membership functions are used for each input corresponding to two regimes (or beliefs) as being perceived by the boundedly rational agents, namely "*low*" and "*high*" in linguistic terms. The reinforcement learning algorithm uses eight distinct states to characterize the underlying financial input variable. Each of the states corresponds to one of the eight (2^3) rules used in the fuzzy setup. For example, the s_1 state corresponds to a rule where all the linguistic terms are "low" for the return and volatility variable, whereas the s_8 state corresponds to the rule with "high" terms for all inputs. Intermediate states relate to other combinations (mixing) of low/high return and volatility inputs. However the state intervals and the resulting membership function parameters estimated for each of the time-series used are different. Thus, the state s_1 for one underlying financial asset may indicate a very different degree of price fluctuations compared with the same observed state for another asset since they are likely to be influenced by different idiosyncratic factors and economic conditions. In summary, the elements of the reinforcement learning algorithm are presented in Table 1.

Formally, let $\mathbf{x}=(x_1,\ldots,x_n)^T$ be an input vector. The fuzzy inference rules for a first-order Sugeno fuzzy system have the following form:

$$R_i : \text{IF } x_1 \text{ is } \mathrm{M}_1^i \text{ AND} \ldots \text{AND } x_n \text{ is } \mathrm{M}_n^i \text{ THEN } y=z^i \quad (\text{Rule } i)$$

where $j=1,\ldots,n$ and $i=1,\ldots,N$, M_j^i is the linguistic label of x_j participating in the ith rule and z^i is the first-order polynomial output of the ith rule. In this study a three-input model is used having the following polynomial output $z^i=h_i+c_ix_1+d_ix_2+k_ix_3$. The number of rules is $N=\prod_{i=1}^{N} N_{M_i}$, where N_{M_i} is the number of fuzzy sets (linguistic labels) of x_j. Consequently, this model comprises two parameter sets, namely the consequent/polynomial parameters and the antecedent/membership function parameters, which are time-varying and adaptively updated in order to account for dynamic persistence and structural changes in the input variables. Since, the conjunction in the premise part of the rules is the product, the truth values of the rules (or rule weights) are calculated as $w_{R_i}=\prod_{j=1}^{n}\mu_{M_j^i}(x_j)$, where $\mu_{M_j^i}(x_j)$ is the membership function associated with M_j^i. The partitions on every input domain are defined by $(\forall x)(\sum_i w_{R_i}(x)=1)$ where w_{R_i} is the truth value of rule i. Alternatively, the structure of the fuzzy rules corresponds to the following representation:

$$\text{IF } \mathbf{x} \text{ is } \Omega_1 \quad \text{THEN } y=z^1$$

$$\ldots$$

$$\text{IF } \mathbf{x} \text{ is } \Omega_N \quad \text{THEN } y=z^N$$

where Ω_i is the premise part of rule i. By definition Ω_i is called *modal vector*, correspondent to rule i, or a *prototype* for rule i. A modal vector has for components the parameters of the membership function selected for each input in the premise part of a rule (Glorennec, 2000). The truth values are normalized $\overline{w}_{R_i}=w_{R_i}/(\sum_{R_i\in A(x)}w_{R_i})$, where $A(\mathbf{x})$ is the set of all rules. After the calculation of the truth values, the total output y_z is estimated as the weighted average of the output of each rule: $y_z=(\sum_{R_i\in A(\mathbf{x})}w_{R_i}z^i)/(\sum_{R_i\in A(\mathbf{x})}w_{R_i})$. The rules R_i correspond to a set of discrete actions. Each action has a "quality factor" i.e., a vector of parameters of the membership functions which is adjusted through learning phase by the Actor. Additionally, each rule has a conclusion part z^i that is adjusted by the Critic in order to estimate the value function. The Critic estimates the policy through fuzzy output over all consequent parts, and the reinforcement signal is calculated. Then based on the reinforcement signal and the update learning algorithm, the conclusion polynomials and action parameters are fine-tuned towards more reinforcement. Knowledge extraction is the most natural characteristic of Actor–Critic learning; several actions are associated with every state, thus several competing conclusions to every rule. The learning process results in

[1] Please see Section 6 for further details on the specification selection process.

determining the best set of rules, videlicet the antecedent and consequent parameters which optimize the future reinforcements.

4.2. Critic details

Let s_t be the input state and $A(\mathbf{x}_t)$ the active rule set at time step t. The value function derived by the output of the fuzzy inference system is the following:

$$\hat{V}_t(s_t) = y_z = \left(\sum_{R_i \in A(\mathbf{x}_t)} w_{R_i}(s_t) z_t^i\right) \Bigg/ \left(\sum_{R_i \in A(\mathbf{x}_t)} w_{R_i}(s_t)\right) \tag{2}$$

If all normalized truth values are denoted by a row vector $\mathbf{W_t}(s_t)$ and all z_t^i by the vector $\mathbf{Z}_t$, the last equation can be reformulated in the following matrix format:

$$\hat{V}_t(s_t) = \mathbf{W}_t \cdot Z_t^T(s_t) \tag{3}$$

Furthermore, if $\mathbf{Z}_t$ is analyzed in first-order polynomials for all z_t^i then input values are also incorporated and thus Eq. (3) is given by

$$\hat{V}_t(s_t) = \mathbf{W}_t \cdot \mathbf{x}^T \cdot \mathbf{P}_t^T = \mathbf{W}_{x_t} \cdot \mathbf{P}_t^T \tag{4}$$

where $\mathbf{P}$ is the vector of the consequent parameters and $\mathbf{W_x}$ the "normalized input" matrix. In further analysis, the above equations (the time index is temporarily dropped to improve readability) have the following form:

$$\begin{aligned}\hat{V}_t(s_t) = y_z &= \overline{w}_1(c_1x_1 + d_1x_2 + k_1x_3 + h_1) + \cdots + \overline{w}_N(c_Nx_1 + d_Nx_2 + k_Nx_3 + h_N) \\ &= [\overline{w}_1x_1 \;\; \overline{w}_1x_2 \;\; \overline{w}_1x_3 \;\; \overline{w}_1 \ldots \overline{w}_Nx_1 \;\; \overline{w}_Nx_2 \;\; \overline{w}_Nx_3 \;\; \overline{w}_N] \cdot [c_1 \;\; d_1 \;\; k_1 \;\; h_1 \;\; \ldots \;\; c_N \;\; d_N \;\; k_N \;\; h_N]^T = \mathbf{W_x} \cdot \mathbf{P}^T\end{aligned} \tag{5}$$

Direct matrix inversion techniques can be used on Eq. (5). The solution for the vector $\mathbf{P}^T$ if the $\mathbf{W_x}$ matrix was invertable, could be

$$\mathbf{P}^T = \mathbf{W}_\mathbf{x}^{-1} \cdot \mathbf{Y} \tag{6}$$

where $\mathbf{Y}$ is the vector of aggregated system output. However, this involves inverting input matrix $\mathbf{W_x}$ which is problematic when columns are dependent, or nearly independent. Direct inverting would mean that there no serial autocorrelation and/or no noise in the input data, which is unrealistic especially in financial applications. Other methods such as triangular or robust orthogonal decomposition are better in terms of under/over-determinacy, but often produce numerical instabilities and result in noise overfitting. In this study the Singular Value Decomposition method (SVD) (Golub and Reinsch, 1971; Golub and van Loan, 1989; Horn and Johnson, 1991) is proposed as a novel approach. The SVD method has the advantage of using principal components to remove unimportant information related to white or heteroscedastic noise and thereby lessens the chance of overfitting. The $\mathbf{W_x}$ matrix is decomposed into a diagonal matrix $\mathbf{D}$ that contains the singular values, a matrix $\mathbf{U}$ of principal components, and an orthogonal normal matrix of right singular values $\mathbf{V}$. The eigenvalues in $\mathbf{D}$ are positive and arranged in decreasing order. If there is heteroscedasticity, the number of significant principal components would be larger than one. Therefore to remove noise, the columns of $\mathbf{U}$ that correspond to small diagonal values in $\mathbf{D}$ are removed. The final vector of the Critic parameters is estimated as follows:

$$\mathbf{P}^T = \mathbf{V} \cdot \mathbf{D}^{-1} \cdot \mathbf{U}^T \cdot \mathbf{Y} \tag{7}$$

After setting the value function, the Critic calculates the reinforcement signal for the next episode. Then, this is used by the Actor in calibrating the optimal antecedent parameters. According to the *direct policy method*, the reinforcement signal is represented by the prediction error, which in this study is the mean squared error, $E_t = 0.5(\hat{V}_t(s_{t+1}) - \hat{V}_t(s_t))^2$. Since the output is the 1-day-ahead projection of the mean-reverting stationary return series, the realized trading value is used as an unbiased estimator $y^o = E[\hat{V}_t(s_{t+1})]$ similarly to a supervised learning paradigm.[2] Thus, the reinforcement signal is given by $E_t = 0.5(y^o - \hat{V}_t(s_t))^2$.

4.3. Actor details

Based upon the state of the environment and the action policy, the Actor part estimates the optimal membership function parameters via an exploration/exploitation mechanism. Technically, two symmetric triangular membership functions are used for each input. This type of fuzzy functions is commonly reported to optimize the training performance

[2] In this setup, the error is perfectly informative i.e., the Critic learning is based on the target value vector. This SVD-based technique prevents Critic from learning oscillations and local sub-optimal solutions as well as increases learning speed for the entire Actor–Critic system. Alternatively, a gradient descent method could be performed on **Z** or **P** directly. However, in combination with the gradient descent algorithm used in the Actor part (see Section 4.3), the resulting two "streams" of optimization algorithms would add a computational overhead, endanger convergence and thus could destabilize the system (Antunović and Cummer, 2004):

in terms of computational load (Ishibuchi et al., 1995). The membership function is defined as follows:

$$\mu_{M_j^i}(x_j) = \begin{cases} 1-(2|x_j-a_j^i|/b_j^i) & \text{if } \ 2|x_j-a_j^i| \le b_j^i \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Moreover, they appear to be well-suited for modeling agent's uncertainty perception mechanism, given that the two parameters, namely a_j^i "peak" and b_j^i the "support" parameter, can implicitly correspond to the "*degree*" and "*range*" of belief. Since the antecedent parameters represent action quality values, a fine-tuning based on *direct policy method* (Sutton et al., 2000; Kimura and Kobayashi, 1998) is implemented. Specifically, the gradient descent update algorithm $\Omega_{t+1}^i = \Omega_t^i - \eta \cdot \nabla E_t$ in modal vector representation is used for each rule. Alternatively, in vector parameter representation as in Eq. (1), the update algorithm is given as follows:

$$\mathbf{v}_{t+1}^i = \mathbf{v}_t^i - \boldsymbol{\eta} \cdot \nabla E_t \quad \forall R_i \in A(x_t) \tag{9}$$

where $\mathbf{v}^i = (a_j^i, b_j^i)^T$ the parameter vector, $\boldsymbol{\eta} = (\eta_a, \eta_b)^T$ the learning rates (e.g., determine the change of parameter values and the convergence of the error function), ∇ the current gradient and E_t the reinforcement signal from the Critic part.[3] The update algorithm provides the exploration/exploitation scheme consisting of an exploiting part $\mathbf{v}_t^i$ and a directed exploring part $\boldsymbol{\eta} \cdot \nabla E_t$ of each particular action. The latter performs an exploration based on learned knowledge and past experience and it is added to the action's quality $\mathbf{v}_t^i$ carrying the current learned policy. The update rule of each parameter is

$$a_{j,t+1}^i = a_{j,t}^i - \eta_a(\partial E_t/\partial a_j^i) \tag{10}$$

$$b_{j,t+1}^i = b_{j,t}^i - \eta_b(\partial E_t/\partial b_j^i) \tag{11}$$

To ensure stability in the learning process, each rate must be less than the reciprocal of the largest eigenvalue of the "normalized" input matrix $\mathbf{W_x}$ (Antunović and Cummer, 2004). The following chain rule is used to analyze the total derivative to its partial derivatives:

$$\partial E_t/\partial a_j^i = (\partial E_t/\partial y_z)(\partial y_z/\partial y_{z^i})(\partial y_{z^i}/\partial w_{R_i})(\partial w_{R_i}/\partial \mu_{M_j^i})(\partial \mu_{M_j^i}/\partial a_j^i) \tag{12}$$

The partial derivatives are derived below:

$$E_t = 0.5(y^o - \hat{V}_t(s_t))^2 = 0.5(y^o - y_z)^2 \Rightarrow \partial E_t/\partial y_z = y^o - y_z = \delta \tag{13}$$

$$y_z = \sum_{i=1}^{N} y_{z^i} \Rightarrow \partial y_z/\partial y_{z^i} = 1 \tag{14}$$

$$y_{z^i} = (w_{R_i} z^i) \Bigg/ \left(\sum_{R_i \in A(\mathbf{x})} w_{R_i} \right) \Rightarrow \partial y_{z^i} \Bigg/ \partial w_{R_i} = (z^i - y_z) / \sum_{i=1}^{N} w_{R_i} \tag{15}$$

$$w_{R_i} = \prod_{j=1}^{n} \mu_{M_j^i}(x_j) \Rightarrow \partial w_{R_i}/\partial \mu_{M_j^i} = w_{R_i}/\mu_{M_j^i} \tag{16}$$

$$\partial \mu_{M_j^i}/\partial a_j^i = \begin{cases} 2\,sign(x_j - a_j^i)/b_j^i, & 2|x_j - a_j^i| \le b_j^i \\ 0, & 2|x_j - a_j^i| > b_j^i \end{cases} \tag{17}$$

$$\partial \mu_{M_j^i}/\partial b_j^i = (1 - \mu_{M_j^i})/b_j^i \tag{18}$$

After chain partial derivation, the error derivatives are analyzed as follows:

$$\partial E_t/\partial a_j^i = [2\delta w_{R_i}(z^i - y_z) sign(x_j - a_j^i)] \Bigg/ \left(b_j^i \mu_{M_j^i} \sum_{i=1}^{N} w_{R_i} \right) \tag{19}$$

$$\partial E_t/\partial b_j^i = [\delta w_{R_i}(z^i - y_z)(1 - \mu_{M_j^i})] \Bigg/ \left(b_j^i \mu_{M_j^i} \sum_{i=1}^{N} w_{R_i} \right) \tag{20}$$

[3] The Hessian can be also used, $\mathbf{v}_{t+1}^i = \mathbf{v}_t^i - \mathbf{H_k^{-1}} \cdot \nabla E_t$, but it is computationally expensive. In this case an approximation $\mathbf{H} = \mathbf{J^T J}$ with $\nabla E_t = \mathbf{J^T} E_t$ where $\mathbf{J}$ the Jacobian, reduces the computational overhead.

where *sign*$(arg)=1$ if *arg* $\succ 0$ and zero otherwise. The adaptive rule for the "*degree-of-belief*" parameter is provided in the following recursive equation:

$$a^i_{j,t+1}=a^i_{j,t}-\eta_a[2\delta w_{R_i}(z^i-y_z)sign(x_j-a^i_j)]\bigg/\left(b^i_j\mu_{M^i_j}\sum_{i=1}^{N}w_{R_i}\right) \tag{21}$$

and for the "*range-of-belief*" parameter:

$$b^i_{j,t+1}=b^i_{j,t}-\eta_b[\delta w_{R_i}(z^i-y_z)(1-\mu_{M^i_j})]\bigg/\left(b^i_j\mu_{M^i_j}\sum_{i=1}^{N}w_{R_i}\right) \tag{22}$$

The learning process could be repeated until there is no significant change in $(a^i_j,b^i_j)^T$ values for each rule (e.g., $|\mathbf{v}^i_{t+1}-\mathbf{v}^i_t|\leq 10^{-3}$) and a convergence goal criterion is met ($|E_{t+1}-E_t|\leq 10^{-5}$). Alternatively, a neural network' training scheme is used in this study. In that a "validation" set containing different observations from the original sample is applied to improve generalization. The validation error normally decreases, as does the in-sample error. However, when the algorithm losses its generalization power, the validation error will begin to rise. When this happens, $\mathbf{v}^i=(a^i_j,b^i_j)^T$ at the minimum of the validation error is returned.

To sum up, in each episode of the adaptive fuzzy Actor–Critic learning algorithm, in the Critic part the polynomial parameters of the value function are calculated using the SVD method, while the membership parameters remain fixed. Thereafter, the fuzzy output (policy) is produced using the previously calculated polynomial parameters and in the Actor part the reward error signal is used to determine the membership parameter updates (action), based on the estimated policy in the previous step. In Fig. 1 the structure of the proposed system is schematically depicted.

4.4. State space: asset returns and volatility dynamics

The reinforcement learning algorithm uses the state space to allocate credit for the learning agent's actions in order to acquire the optimal selection policy. In this study the state is used to track the time-varying characteristics of the underlying time-series. In financial modeling, the geometric Brownian motion is the standard model of price growth over time (Hull, 2000). Thus, the return of the underlying financial variable is described in the following equation:

$$Y_{t+\Delta t}/Y_t\sim N(\mu(t)\Delta t,\sigma(t)\sqrt{\Delta t}) \tag{23}$$

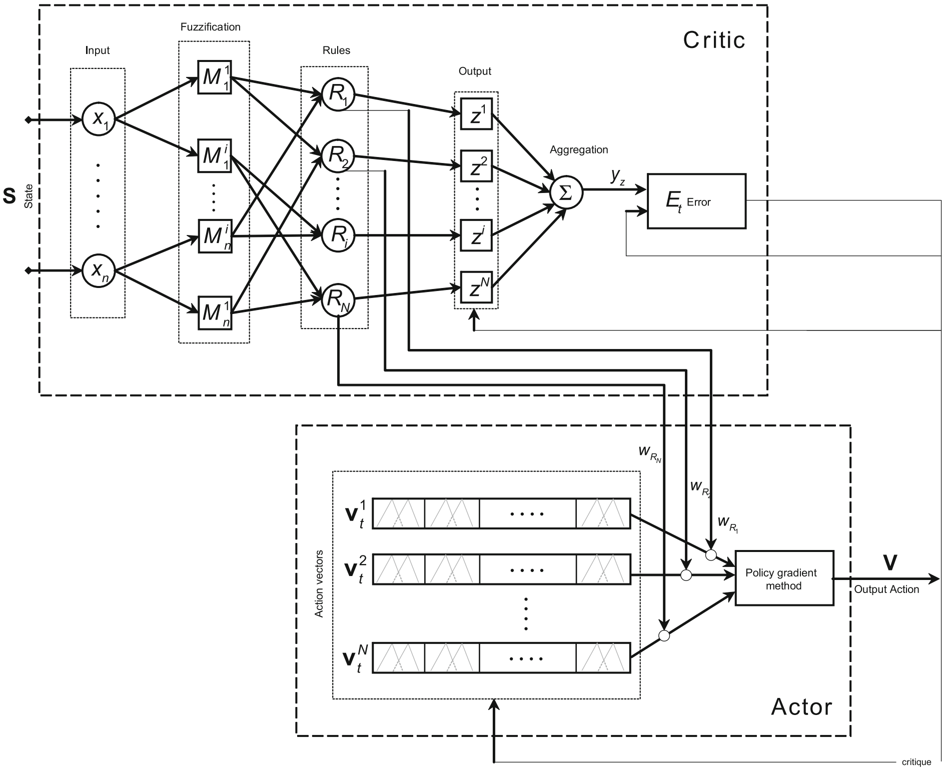

Fig. 1. The adaptive fuzzy Actor–Critic reinforcement learning system.

where Y is the price and N the normal distribution with expected mean $\mu(t)$ and standard deviation $\sigma(t)$. In active trading, the expected rate of return indicating the direction of the asset's movement is a critical determinant in making decisions.

Moreover, in order to assess the time-varying dynamics of the underlying variable technical analysts depend usually on historical standard deviation. However, fluctuations in actual prices, beyond historical standard deviation and many times even greater than those implied by changes in the market fundamentals, are inferred by Shiller (1989) as being the result of waves of optimistic or pessimistic market psychology. The sharp stock US market decline of 22% on October 19, 1987, in the complete absence of news about fundamentals, appears to contradict the Efficient Markets Hypothesis. Specifically, according to the "time-varying risk premium theory" (Bekaert and Wu, 2000) the return shocks are caused by changes in *conditional volatility*. When 'bad' news arrives in the market the current volatility increases and this causes upward revisions of the conditional volatility. This increased conditional volatility has to be compensated by a higher expected return, leading to an immediate decline in the current value of the market. While bad news generates an increase in volatility, the net impact of good news in not clear. Another illustration of the linkage—in this case asymmetric-between conditional volatility and expected returns may be offered through the "leverage effect" (Christie, 1982). A negative (positive) return increases (reduces) financial leverage, which makes the stock riskier (less risky) and increases (reduces) volatility. The causality however here is different: the return shocks lead to changes in conditional volatility, whereas the time-varying premium theory contends the opposite. An alternative rationalization for the relation may be offered by invoking "trigger strategies" in the equity markets (Krugman, 1987). Institutional participants in equity markets react whenever the maximum expected loss of portfolios, as measured for example by the Value-at-Risk (VaR), reaches a predetermined level and therefore share price dynamics are being driven, partly, by revisions in the measured conditional volatility.[4] Each time the conditional volatility rises, a number of those portfolios deviate from their pre-determined level of VaR, hitting their risk limits and this generates a re-allocation of assets towards safer ones. When portfolio insurers leave the market the stock prices must fall in order for the other investors to be given an incentive to hold a larger quantity of stock.

Recently, many researchers also studied the contemporaneous relationship between the stock returns and the associated changes in the level of implied volatility indices (Whaley, 2000; Simon, 2003). Specifically, for the S&P100 index and the implied volatility VIX index, this relationship has been found to be asymmetric in the sense that negative stock index returns are associated with greater proportional changes in implied volatility measures than are positive returns. The explanation offered is that option traders react to negative returns by bidding up the implied volatility. Nonetheless, empirically there is a growing debate whether the implied volatility can be used as a forward indicator of the underlying equity index. This issue has not been treated properly in the literature with the exception of a study by Giot (2005), in which he concludes that positive forward-looking returns are to be expected for long positions at high levels of the implied volatility indices.

Finally, it is investigated whether a nonlinear functional form exists and also its consequent predictability. Christoffersen and Diebold (2006) show that volatility dependence produces return dependence, and therefore forecastability, as long as expected returns are nonzero. The intuition behind this relationship is that *volatility changes* will alter the probability of observing negative or positive returns. Specifically, the "higher" the volatility, the "higher" the probability of a negative return as long as the expected returns are positive. In that context, the predictive return sign ability of trading rules that rely on a simple switching policy is examined: positive predicted returns are executed as long positions and negative returns as short positions. A similar policy has been employed, with considerable success, by a number of other researchers (Gençay, 1998a, 1998b; Fernández-Rodriguez et al., 2000) buy/sell signals are produced from technical trading rules that incorporate various linear or nonlinear econometric models. In general terms they find that the profitability of this "active" policy is higher than from a "passive" one including transaction costs.

As a result, although it is widely accepted in theory and through empirical evidence that expected return and conditional volatility determine the time-varying behavior of financial assets, the exact relationship and causality is not generally agreed. This "fuzzy interaction" is taken into consideration in the reinforcement learning algorithm by means of inference rules, corresponding to the rules-of-thumb used by technical analysts. Consequently, the fuzzy rules comprise the state space based on which the learning agent allocates credit for the actions acquired.

In this study various conditional volatility models are used in the input and state space. The basic assumption in these models is that daily returns follow the stochastic process $y_t = \mu_t + e_t = \mu_t + \sigma_t \varepsilon_t$ with conditional distribution function $F_t(\cdot)\mu_t = E(y_t|F_{t-1})$, $\sigma_t^2 = E(e_t^2|F_{t-1})$ and $\varepsilon_t = (e_t/\sigma_t)$. Under parametric approaches specific distributions for $F_t(\cdot)$ are considered, such as the Gaussian $N(0,1)$, the Student-t or the Generalized Error Distribution (*GED*). The moving average model of Alexander (1998) is given by the equation

$$\sigma_t^2 = [1/(\kappa-1)] \sum_{j=1}^{\kappa} (y_{t-j} - \mu_t^{\kappa})^2 \tag{24}$$

where $\mu_t^{\kappa} = (1/\kappa)\sum_{j=1}^{\kappa} y_{t-j}$ is the moving average of κ days. Alternatively, conditional variance can be estimated by one of the family of *GARCH* models (Bollerslev, 1986). In particular, the GARCH(1,1) model is given by

$$\sigma_t^2 = \omega_0 + \omega_1 (y_{t-1} - \mu_t)^2 + \omega_3 \sigma_{t-1}^2 \tag{25}$$

[4] VaR depends entirely on a multiple of the estimated conditional volatility under the assumption of normally distributed returns.

A special case of GARCH models is the Exponentially Weighted Moving Average (*EWMA*) specification, adopted by the *Riskmetrics* (*RM*) model of J.P. Morgan, under which

$$\sigma_t^2 = \lambda\sigma_{t-1}^2 + (1-\lambda)(y_{t-1}-\mu_t)^2 \tag{26}$$

Riskmetrics has chosen $\lambda = 0.94$ and 0.97 as the optimal decay factor for daily and monthly data, respectively (Jorion, 2000).

5. Other competing models

The adaptive fuzzy Actor–Critic system is compared against a Markov-switching model and a Recurrent Neural Network, in terms of relative predictability and profitability performance. These models are described below.

5.1. Markov-switching model

A well-established class of regime-switching models assumes that the regime that occurs at time t cannot be observed and is determined by an unobservable process or state, denoted as s_t. In case of only two regimes (e.g., "low" and "high") and AR(1) specification in both regimes, the model is given by

$$y_t = \begin{cases} \varphi_{0,1} + \varphi_{1,1} y_{t-1} + \varepsilon_t & \text{if } s_t = 1 \\ \varphi_{0,2} + \varphi_{1,2} y_{t-1} + \varepsilon_t & \text{if } s_t = 2 \end{cases} \tag{27}$$

The most popular model in this class is the Markov-switching model introduced by Hamilton (1989). In that s_t is assumed to be a first order Markov-process, i.e., s_t depends only on s_{t-1}. The transition probabilities are $P(s_t=1|s_{t-1}=1)=p_{11}$, $P(s_t=2|s_{t-1}=1)=p_{12}$, $P(s_t=1|s_{t-1}=2)=p_{21}$ and $P(s_t=2|s_{t-1}=2)=p_{22}$ with $p_{11}+p_{12}=1$ and $p_{21}+p_{22}=1$. Additionally, via the theory of ergodic Markov chains it can be shown (Hamilton, 1994) that for the two-state model, the unconditional probabilities are given by $P(s_t=1)=(1-p_{22})/(2-p_{11}-p_{22})$ and $P(s_t=2)=(1-p_{11})/(2-p_{11}-p_{22})$.

5.2. Recurrent neural network

A single hidden layer feedforward network with sufficiently hidden units (neurons) and properly adjusted parameters can theoretically approximate any function to any desired degree of accuracy.[5] The output of a neural network is produced via the application of a transfer function. The functionality is to modulate the output space as well as prevent outputs from reaching very large values which can "block" training.[6] Learning typically occurs through training, where the training algorithm iteratively adjusts the connection weights. Common practice is to divide the sample into three distinct sets namely, training, validation and testing (out-of-sample) set; the training set is the largest and is used to learn the patterns presented in the data, the validation set is used to evaluate the generalization ability in order to avoid overfitting and the training set should consist of the most recent observations that are processed for testing predictability.[7] Specifically, if $\mathbf{x} = (x_1, \ldots, x_p)^T$ is the input vector of a single layer feedforward network with q hidden units, the output is given by

$$y_t = K\left[\beta_0 + \sum_{i=1}^{q} \beta_i G\left(\gamma_{i0} + \sum_{j=1}^{p} \gamma_{ij} x_{j,t}\right)\right] = f(\mathbf{x_t}, \mathbf{z}) \tag{28}$$

where $i = 1, \ldots, q$ and $j = 1, \ldots, p$. Let $\mathbf{u} = (\beta_0, \ldots, \beta_q, \gamma_{11}, \ldots \gamma_{ij} \ldots, \gamma_{qp})^T$ be the weight vector and K, G the transfer functions. The solution of the network considers estimation of the unknown vector $\mathbf{u}$ with a sample of data values. A recursive estimation methodology, which is called backpropagation is used to estimate the weight vector, as follows:

$$\mathbf{u}_{t+1} = \mathbf{u}_t + \eta \cdot \nabla f(\mathbf{x_t}, \mathbf{u}_t) \cdot [y_t - f(\mathbf{x_t}, \mathbf{u}_t)] \tag{29}$$

where $\nabla f(\mathbf{x_t}, \mathbf{u})$ is the gradient vector with respect to $\mathbf{u}$ and η the learning rate. The learning rate modulates the size of the change of the weight vector on the tth iteration. The $\mathbf{u}$ vector update is achieved usually via the minimization of the mean squared error function.

While feedforward neural networks appear to have no memory since the output at any time-instant depends entirely on the inputs and the weights at that instant, recurrent neural networks exhibit characteristics that simulate short-term

[5] Despite the importance of selecting the optimum number of hidden neurons, there is no explicit formula for that matter. The geometric pyramid rule proposed by Masters (1993) considers $\sqrt{pm}$ neurons for a three-layer network with p inputs and m outputs. Katz (1992) indicates that an optimal number of hidden neurons can be found between one-half to three times the number of inputs, whereas Ersoy (1990) proposes doubling the number of neurons until the network's RMSE performance deteriorates.

[6] Levich and Thomas (1993) and Kao and Ma (1992) found that hyperbolic sigmoid and logistic transfer functions are appropriate for financial markets data because they are nonlinear and continuously differentiable which are desirable properties for network learning.

[7] The validation error starts decreasing until the network begins to overfit the data and the error will then begins to rise. The weights are calculated at the minimum value of the validation error.

memory. In this study, Elman Recurrent neural networks (Elman, 1990) have been utilized. In Elman networks with a single hidden layer, the lagged outputs of the hidden neurons are fed back into the hidden neurons themselves. If $\mathbf{x_t}$ is the input vector with q hidden units and t the time index, the output of the network is given by

$$y_t = G\left[\beta_0 + \sum_{i=1}^{q} \beta_i G\left(\gamma_{i0} + \sum_{j=1}^{p} \gamma_{ij} x_{j,t} + \sum_{h=1}^{q} \delta_{ih} g_{h,t-1}\right)\right] + \varepsilon_t \tag{30}$$

where

$$g_{i,t} = G\left(\gamma_{i0} + \sum_{j=1}^{p} \gamma_{ij} x_{j,t} + \sum_{h=1}^{q} \delta_{ih} g_{h,t-1}\right) \tag{31}$$

The weight vector is $\mathbf{u} = (\beta_0, \ldots, \beta_q, \gamma_{11}, \ldots \gamma_{ij} \ldots, \gamma_{qp}, \delta_{11}, \ldots \delta_{ih} \ldots, \delta_{qq})^T$ and G is the logistic transfer function.

6. Empirical results

The predictability of the Adaptive Fuzzy Actor–Critic Learning System (denoted AFACL) is examined against that of a Recurrent Neural Network (RNN) and a Markov-switching model (MSW), while a Buy and Hold (B&H) strategy is also used as benchmark. The sample comprises daily logarithmic returns of NASDAQ Composite (USA), NIKKEI255 (Asia) and FTSE100 (Europe) indices, from 1/2/1984 to 7/14/2009 (6662 observations). The out-of-sample performance is investigated in two disjoint 4-year periods, 4/8/1998–2/5/2002 (1000 obs.) and 9/14/2005–7/14/2009 (1000 obs.) with the use of a 1-day rolling window. To enhance robustness in the results, both out-of-sample periods are further segmented into expanding sub-periods. The final backtesting periods are P_1: 4/8/1998–2/20/2001, P_2:4/8/1998–2/5/2002, P_3: 9/14/2005–7/29/2008 and P_4: 9/14/2005–7/14/2009. The total period is particularly interesting for technical analysts as it contains very turbulent intervals, diverse regimes and several “extreme” events including the Asian crisis, the rise and fall of the tech-market bubble and the financial crisis of 2008–2009, the latter of which lead to global recession and was caused by the credit insolvency of investment institutions.

The input space of the AFACL system comprises two lagged returns and one lag of the conditional volatility daily changes $\Delta\sigma = \sigma_t - \sigma_{t-1}$. The input selection is conducted according to the methodology of Franses and van Dijk (2000) on nonlinear models and neural networks.[8] The output is the forecasted 1-day-ahead return $\hat{y}_t$. The conditional volatility daily changes are calculated via a 20-day moving average model, an exponentially weighted moving average model with 0.94 decay factor and GARCH(1,1). The corresponding notation is AFACL-MA(20), AFACL-RM(0.94)[9] and AFACL-GARCH(1,1). Based on the same methodology as in AFACL, the RNN model uses two lagged returns, a topology comprising 10 g neurons in the hidden layer and a single-output layer y.[10,11] The MSW model uses an AR(2) specification in both regimes as well as two regimes/beliefs (i.e., “low” and “high”) for the state variable, in direct association with the architecture of the AFACL system.

For the out-of-sample testing period the models utilize a rolling window of all previous observations as a training sample and produce forecasts for each day within the corresponding period. In case of AFACL and RNN the validation sample for each period is the 25% of the training set, and is used to evaluate the generalization ability and avoid overfitting. The training set consists of the most recent observations that are processed in each period. The training and validation samples utilize a moving window of all previous observations in order to produce forecasts for each day within each backtesting period. The process is repeated in each of the expanding periods.

In order to account for the use of nonlinear models a test for the presence of nonlinear dependence in the series is conducted. In that, the well-known BDS test statistic is used, which under the null of i.i.d. is given by (Brock et al., 1991):

$$W_{m,T}(\varepsilon) = T^{1/2}[C_{m,T}(\varepsilon) - C_{1,T}^{m}(\varepsilon)]/\sigma_{m,T}(\varepsilon) \tag{32}$$

$C_{m,T}(\varepsilon)$ is the correlation integral from m dimensional vectors that are within a distance ε from each other, when the total sample is T, and $\sigma_{m,T}(\varepsilon)$ is the standard deviation of $C_{m,T}(\varepsilon)$. Under the null hypothesis, $W_{m,T}(\varepsilon)$ has a limiting standard normal distribution. The BDS test is applied on (a) the original data, (b) the residuals from an autoregressive filter AR(2)—based on the selected return lags—in order to ensure that the null is not rejected due to linear dependence and (c)

[8] Specifically, the procedure for the selection of the lags involved in the first step the calculation of the Ljung–Box statistics for the first 10 lags of all return series in order to get a first indication. Significant autocorrelations of up to the second lag of the return series were identified. Additionally, the Akaike and Schwarz Information Criteria (AIC, SIC) that were estimated for the first six lags provided the minimum value at the second lag. Moreover, a sensitivity analysis based on RMSE conducted stepwise on the AFACL system (in case of a RNN model only for return lags), which involved three lags of the conditional volatility changes and six lags for returns, revealed that the selected setup provided with the best results. All other topologies were found to be qualitatively worse or roughly equal compared with those finally presented in Table 3 (i.e., for the AFACL system a topology with three return lags and one lag of the volatility changes provides with roughly equal results, albeit with computational overhead due to the added input variable). The results of the sensitivity analysis are available upon request.

[9] The exponentially moving average corresponds to the approach adopted by *RiskMetrics* (J.P. Morgan) and for that reason it is denoted here as RM (0.94).

[10] Please refer to footnote 8.

[11] This empirical result also follows Katz (1992) and Ersoy (1990).

Table 2
BDS test.

Index	Correlation dim.	m=2				m=3				m=4			
	Dim. distance	ε=1		ε=1.5		ε=1		ε=1.5		ε=1		ε=1.5	
	Total period	PI	PII	PI	PII	PI	PII	PI	PII	PI	PII	PI	PII
NASDAQ Composite	Raw data	37.671	25.153	36.145	25.181	45.860	35.154	44.472	34.049	51.296	42.353	48.581	39.260
	AFR	34.736	25.059	34.449	25.036	42.998	34.951	43.019	33.833	48.344	42.132	47.151	39.054
	NLSSR	13.433	9.915	10.892	8.217	15.011	12.338	12.514	9.795	15.114	13.016	12.638	10.292
FTSE100	Raw data	9.930	16.985	11.916	19.886	13.000	22.332	15.373	25.442	15.186	26.525	17.434	29.097
	AFR	9.785	16.607	11.871	19.571	12.665	21.855	15.159	25.042	14.817	26.080	17.216	28.750
	NLSSR	5.517	5.981	5.338	5.597	5.108	6.252	4.917	5.982	4.348	5.933	4.157	5.620
NIKKEI255	Raw data	24.704	14.725	23.491	14.882	33.338	21.097	30.759	20.312	39.566	26.233	35.119	23.934
	AFR	24.513	14.081	23.237	14.331	33.289	20.741	30.615	19.950	39.568	25.990	35.021	23.669
	NLSSR	11.618	6.931	10.363	6.149	13.657	8.046	12.042	6.944	13.983	8.827	12.035	7.363

Notation: Raw data=daily index returns, AFR=residuals from an autoregressive filter AR(2), NLSSR=natural logarithm of the squared standardized residuals from the AR(2)-GARCH-M (1,1) model; m=dimension, ε=number of standard deviations of the data; Significance at the 1% level corresponds to the critical value 2.58; PI: 1/2/1984–2/5/2002, PII: 1/2/1984–7/14/2009.

the natural logarithm of the squared standardized residuals from an AR(2)-GARCH-M(1,1) model in order to ensure that rejection of the null is not due to conditional heteroscedasticity (De Lima, 1996).

In all three cases the null of i.i.d. at the 1% marginal significance level could be rejected and the evidence seemed to suggest that a genuine nonlinear dependence is present in the data (Table 2).

The trading strategy of the technical analyst works as follows; at the end of each trading day the models are being re-estimated over a rolling sample with a length equal to the training period. When the output of a model is greater than zero this is used as a buy signal and a value less than zero as a sell signal. The total return, when transaction costs are not considered, is estimated as

$$W=\sum_{t=1}^{T_o}\theta_t y_t \tag{33}$$

where T_o indicates the out-of-sample horizon, y_t is the realized return and θ_t is the recommended position which takes the value of (-1) for a short and $(+1)$ for a long position (e.g., Gençay, 1998b; Jasic and Wood, 2004). In order to evaluate the forecast accuracy of the models, the percentage of correct predictions or correctly predicted signs was calculated as follows:

$$\text{Sign rate}=H/T_o \tag{34}$$

where H is the number of correct predictions. Two other comparative profitability measures were also considered: the Ideal Profit (IP) and the Sharpe ratio (SR). The IP compares the system return against the perfect forecaster and is calculated by

$$IP=\left(\sum_{t=1}^{T_o}\theta_t r_t\right)\Bigg/\left(\sum_{t=1}^{T_o}|r_t|\right) \tag{35}$$

where the value $IP=0$ is considered as a benchmark to evaluate the performance of a trading strategy. When the direction indicator θ_t takes the correct position for all observations in the sample, then $IP=1$, whereas if all forecasted positions are wrong $IP=-1$. The SR is the proportion of the mean return of the trading strategy over its standard deviation. The higher the SR is, the higher the return and the lower the volatility:

$$SR=\mu_{T_o}/\sigma_{T_o} \tag{36}$$

Finally, as a measure of predictability the Henriksson–Merton (HM) statistic (Henriksson and Merton, 1981) was employed for the y_t (realized) and $\hat{y}_t$ (forecasted) returns. The statistic is based on the following contingency table:

$$\begin{array}{ccc} & y_t>0 & y_t\le 0 \\ \hat{y}_t>0 & \left[\begin{array}{c} v_1 \\ \Lambda_1-v_1 \end{array}\right. & \left.\begin{array}{c} v \\ \Lambda_2-v_2 \end{array}\right] \\ \hat{y}_t\le 0 & & \end{array} \tag{37}$$

where v_1 is the number of correct forecasts in "upward" markets, v_2 the number of incorrect forecasts in "downward" markets and Λ_1, Λ_2 the number of "up-market" and "down-market" periods, respectively, in the sample. Henriksson and Merton (1981) showed that v_1 has a hypergeometric distribution under the null hypothesis of no market-timing ability, which may be approximated by

$$v_1\sim N(v\cdot\Lambda_1/\Lambda,\quad [v_1\cdot\Lambda_1\cdot\Lambda_2\cdot(\Lambda-v)]/[\Lambda^2\cdot(\Lambda-1)]) \tag{38}$$

where $\Lambda=\Lambda_1+\Lambda_2$ and $v=v_1+v_2$.

Table 3
Out-of-sample performance of the trading models.

Index	**Model**	**Total Return (%)**				**B&H Return (%)**				**Sign Rate**				**HM test**				**RMSE**				**Sharpe Ratio** (ann.)				**Ideal Profit**			
	Period	P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4	P_1	P_2	P_3	P_4
NASDAQ Composite	AFACL-RM(0.94)	**74.7** (59.0)	**60.9** (39.3)	**22.8** (14.8)	**17.6** (4.0)	24.9	4.5	7.2	-17.7	0.543	0.528	0.522	0.510	2.375***	2.041**	1.258	1.531*	0.024	0.030	0.011	0.017	0.673	0.409	0.442	0.163	0.057	0.034	0.037	0.015
	AFACL-MA(20)	**138.6** (121.7)	**160.3** (136.9)	**12.8** (7.6)	**47.1** (35.9)					0.544	0.538	0.521	0.508	2.580***	2.826***	1.302*	1.590*	0.027	0.026	0.011	0.017	1.252	1.080	0.247	0.436	0.105	0.091	0.022	0.042
	AFACL - GARCH(1,1)	**64.5** (48.3)	**46.1** (25.8)	**23.7** (18.6)	**64.8** (54.3)					0.537	0.524	0.525	0.517	2.154**	1.734**	1.785**	2.002**	0.027	0.026	0.011	0.017	0.581	0.310	0.459	0.602	0.049	0.026	0.039	0.058
	RNN	**39.9** (20.4)	**40.3** (14.6)	**-28.9** (-48.0)	**4.8** (-21.4)					0.528	0.521	0.462	0.474	1.893**	1.850**	-1.397	-0.741	0.029	0.028	0.014	0.021	0.359	0.270	-0.561	0.044	0.030	0.023	-0.048	0.004
	MSW	**12.3** (-4.5)	**36.2** (13.5)	**-27.6** (-41.6)	**-147.1** (-168.6)					0.526	0.517	0.495	0.480	1.240	0.971	-0.928	-1.780	0.024	0.025	0.011	0.017	0.111	0.243	-0.534	-1.369	0.009	0.020	-0.045	-0.132
FTSE100	AFACL-RM(0.94)	**39.8** (21.3)	**64.7** (39.8)	**18.9** (1.8)	**38.4** (15.6)	-1.2	-17.3	-0.5	-23.3	0.518	0.520	0.493	0.494	2.250**	3.191***	1.434*	1.612*	0.012	0.013	0.011	0.018	0.690	0.816	0.378	0.398	0.057	0.068	0.033	0.038
	AFACL-MA(20)	**67.3** (50.0)	**69.7** (46.3)	**62.1** (55.8)	**74.1** (62.4)					0.519	0.518	0.525	0.515	1.922**	2.754***	3.685***	3.001***	0.012	0.013	0.011	0.015	1.169	0.879	1.243	0.770	0.096	0.073	0.109	0.075
	AFACL - GARCH(1,1)	**65.1** (49.5)	**89.7** (67.7)	**15.7** (1.6)	**26.8** (7.4)					0.533	0.527	0.494	0.498	3.223***	3.624***	0.921	1.295*	0.012	0.013	0.011	0.018	1.130	1.133	0.313	0.278	0.093	0.094	0.026	0.027
	RNN	**-76.2** (-95.3)	**-99.0** (-124.7)	**4.3** (-14.0)	**-32.0** (-57.1)					0.468	0.465	0.482	0.479	-1.104	-1.505	-0.031	-0.488	0.015	0.016	0.012	0.019	-1.326	-1.252	0.086	-0.332	-0.109	-0.104	0.007	-0.032
	MSW	**30.7** (12.0)	**58.4** (33.0)	**-56.2** (-75.6)	**-2.3** (-29.1)					0.510	0.520	0.462	0.484	1.450*	2.580***	-2.741	-0.477	0.012	0.013	0.011	0.016	0.533	0.737	-1.125	-0.024	0.044	0.061	-0.099	-0.002
NIKKEI 255	AFACL-RM(0.94)	**11.4** (2.1)	**90.4** (71.9)	**37.6** (26.2)	**25.8** (8.8)	-21.2	-53.2	2.1	-32.6	0.497	0.519	0.488	0.490	1.590*	3.845***	2.438***	2.374***	0.014	0.015	0.013	0.022	0.169	0.933	0.598	0.219	0.015	0.081	0.052	0.023
	AFACL-MA(20)	**35.3** (22.3)	**113.5** (95.3)	**33.1** (19.8)	**27.3** (7.7)					0.503	0.525	0.476	0.471	2.159**	4.488***	1.058	1.288*	0.014	0.015	0.014	0.021	0.509	1.172	0.526	0.231	0.044	0.101	0.046	0.022
	AFACL - GARCH(1,1)	**7.8** (0.2)	**33.8** (20.2)	**10.1** (2.3)	**19.3** (4.5)					0.476	0.480	0.480	0.484	0.991	1.469*	1.285*	1.519*	0.014	0.015	0.013	0.022	0.085	0.347	0.164	0.164	0.007	0.031	0.012	0.016
	RNN	**-23.1** (-41.9)	**-55.3** (-80.2)	**-17.4** (-34.9)	**24.5** (0.4)					0.448	0.441	0.464	0.454	-1.698	-2.660	-0.554	-1.103	0.017	0.018	0.016	0.023	-0.344	-0.611	-0.277	0.208	-0.029	-0.053	-0.024	0.019
	MSW	**5.1** (-18.7)	**12.8** (-17.7)	**1.9** (-20.8)	**20.4** (-7.8)					0.468	0.478	0.457	0.461	-0.259	0.663	-0.809	-0.577	0.014	0.015	0.013	0.019	0.077	0.132	0.031	0.182	0.006	0.011	0.002	0.017

Notation: AFACL=adaptive fuzzy Actor–Critic reinforcement learning system. RNN=recurrent neural network. MSW=Markov-switching model; MA(20)=Moving average with a 20 days window, RM(0.94)=RiskMetrics' exponentially weighted MA rule (decay factor=0.94), GARCH(1,1)= Bollerlsev-GARCH model.HT test=Henriksson and Merton (1981) test, asymptotically distributed as *N*(0,1); In parenthesis total return after transaction costs (0.05% average fixed cost for each one-way trade); The sign rate measures the proportion of correctly predicted signs. The Sharpe ratio is defined as the ratio of the mean return of the strategy over its standard deviation (it has been annualized by multiplying it with the squared root of 250). The ideal profit is the ratio of the returns of the trading strategy over the returns of a perfect predictor; (***), (**) and (*) indicate significance at the one sided 1%, 5% and 10% levels; P_1: 4/8/1998–2/20/2001, P_2:4/8/1998-2/5/2002, P_3: 9/14/2005-7/29/2008, P_4: 9/14/2005-7/14/2009.

The empirical results are reported in Table 3. Considering total returns, the AFACL system under any volatility structure dominates the RNN, MSW and the B&H strategy consistently for all indices in all periods. Specifically, the total profitability of the AFACL system ranges from a minimum of 7.8% (AFACL with GARCH(1,1) volatility in P_1 for NIKKEI255) to a maximum of 160.3% (in P_2 the AFACL-MA(20) for the NASDAQ index). The RNN return varies from -99.0% (FTSE100 in P_2) to 40.3% (in period P_2 for NASDAQ) and that of MSW from -147.1% (NASDAQ in P_4) to 58.4% (in P_2 for the FTSE100 index). The same picture emerges with the inclusion of transaction costs, which are estimated as 0.05% for each one-way trade, following Hsu and Kuan (2005) and Fama and Blume (1966). Again, the AFACL system remains significantly profitable and by far better compared with other models.[12] The proportion of correctly predicted signs for AFACL is also on average higher vis-à-vis the competing models. The significant profitability of AFACL may be compromised with the marginal improvement of the sign rate (in some cases around 50%), yet it is due to the substantial improvement of the quantitative importance of the correctly forecasted signs. The HM test provides a further validation of the statistical significance of the sign rate in case of the AFACL system with highly significant values at the one-sided 1% level, such as 4.488 (AFACL-MA(20) in P_2 for NIKKEI255), 3.191 (AFACL-RM(0.94) for FTSE100 in P_2) or 2.375 (AFACL-RM(0.94) in P_1 for the NASDAQ index). Instead, the RNN or MSW model achieves the former its highest only twice in case of NASDAQ in P_1 and P_2 at the 1% level, and the latter in case of FTSE100 in P_2, while both report many non-significant or negative values. Additionally, the SR measuring the profitability per unit of risk and the IP, are much higher compared with RNN and MSW in all indices and periods examined. The fact that B&H strategy outperforms the RNN model in some cases is not in accordance with previous results derived by Fernández-Rodriguez et al. (2000) as well as with the conclusions reached by Christoffersen and Diebold (2006). It is noticeable in this study that in some cases an "active" trading strategy employed by the nonlinear RNN and MSW models compared with the naive B&H, provides with worse even negative (loss) results, (e.g., in P_3 for both RNN and MSW in case of NASDAQ). However, the B&H strategy never outperforms AFACL system in terms of profitability.[13]

The comparison between the three AFACL volatility specifications shows that simple equally weighted or exponentially weighted moving average models, can produce sign forecasts that are not significantly worse than those obtained from more complicated econometric models such as the GARCH(1,1). However, this is not surprising on the basis of recent empirical literature. For instance, it is documented that for the NASDAQ index, volatility forecasts from moving average rules closely approximate those from GARCH (1,1) models (Schwert, 2002). Furthermore, Simon (2003) also reports that in case of NASDAQ100 the volatility forecasts based on the Glosten–Jagannathan–Runkle GARCH model (Glosten et al., 1993) were found to be 3.0 percentage points higher that the actual, when the exponentially weighted moving average volatility forecasts are only 1.5 percentage points below actual volatility.

Overall, the predictive ability of the AFACL system is significantly higher compared with the other models. It seems plausible that a B&H strategy would be the best in the extreme case of a pure trending market or in absence of turning points in price movement, both of which would imply homogeneity and rationality in trader behavior. However, when there is uncertainty, turbulence and eventually heterogeneity caused by a number of factors that may affect market microstructure, the AFACL will be better in terms of prediction performance.

7. Conclusions

This study addresses the learning issue for boundedly rational agents in the dynamic and noisy environment of financial markets. The main objective is the development of a learning algorithm that "decodes" the knowledge acquisition mechanism and the decision-making process of technical analysts. In this direction a methodology combining reinforcement learning and fuzzy inference is employed. Specifically, a new adaptive fuzzy Actor–Critic reinforcement learning system is developed. The resulting functionality leads to the derivation of the parameters of fuzzy inference rules by means of adaptive training.

The proposed setup advances the literature on heterogeneous learning in speculative markets by introducing a trading system wherein market environment and agent beliefs are represented by fuzzy inference rules. In technical terms, it expands the literature that has utilized Actor–Critic reinforcement learning and fuzzy systems in agent-based applications, by presenting an adaptive fuzzy reinforcement learning approach that results in superior predictions upon the direction-of-change of the market. The purpose of this paper was to illustrate this concretely through a comparative investigation against other well-established nonlinear models. Beyond the existing practice that has utilized return lags and moving

[12] There are only two cases where the return of another model (RNN or MSW) is higher than that of AFACL, but without the inclusion of transaction costs. Specifically in P_4 for the NIKKEI255 index, the return of the MSW and RNN is 20.4% and 24.5%, respectively, compared with 19.3% of the AFACL-GARCH(1,1) system. However, after transaction costs the return of MSW drops to -7.8% (loss) and that of RNN to 0.4%, while the AFACL-GARCH(1,1) system retains a gain of 4.5%.

[13] An anonymous referee suggested further to triangular, the use of trapezoidal or Gaussian membership functions, in a comparative exercise to enhance robustness of the results. In AFACL system two symmetric triangular membership functions are used for each input, since they are known to optimize the training performance in terms of computational load (Ishibuchi et al., 1995). To enhance robustness trapezoidal and gaussian functions were also used indicatively for AFACL-GARCH(1,1) setup in the two disjoint 4-year periods P_2 and P_4 and for all indices. The results indicated that computational time increased on average 20.9% for trapezoidal membership functions and 5.2% for Gaussian, while the quality of results slightly deteriorated in terms of profitability (on average 6.6% and 3.9%, respectively) and statistical significance. The RMSE remained unchanged and the sign rate -following profitability—decreased slightly or remained stable.

averages as input variables and in accordance with financial theory (price growth model), it is demonstrated that the new system leads to enhanced forecastability via the incorporation of conditional volatility. Although it is widely accepted both empirically and theoretically that the expected return and conditional volatility describe the time-varying behavior of asset prices and therefore "reflect" the prevailing conditions of the market environment, their exact relationship is not generally agreed. This "fuzzy interaction" is taken into consideration in the reinforcement learning algorithm by means of inference rules, corresponding to the rules-of-thumb or heuristics used by technical analysts. Consequently, the fuzzy rules comprise the state space based on which the learning agent allocates credit for the actions acquired. Following that, depending on the current state, an optimal parameter vector representing the response (action) to the environment is selected by the Actor–Critic learning agent. The selected Actor and Critic parameters are then used to generate the policy and thus the two-way "directional" trading decision (buy or sell). The policy is represented by the nonlinear output function of the Sugeno fuzzy system, according to the *direct policy approach* coined by Sutton et al. (2000). Subsequently, the quality of the calibrated parameter values (action) is evaluated by the learning agent via the prediction accuracy of the corresponding fuzzy model and specifically by computing the forecasting error, which is used as the reinforcement signal.

Overall, the predictive ability of the adaptive fuzzy Actor–Critic reinforcement learning system is significantly higher compared with the other models. The results indicate that with the inclusion of transaction costs the profitability of the proposed system, in case of NASDAQ Composite, FTSE100 and NIKKEI255 indices, is consistently superior to that of a Recurrent Neural Network, a Markov-switching model and a Buy & Hold strategy. A possible explanation is that a B&H strategy would be the best in the extreme case of a pure trending market or in absence of turning points in price movement, both of which would imply homogeneity in trader behavior. However, when there is uncertainty, turbulence and eventually heterogeneity caused by a number of factors that may affect market microstructure, the novel model will be better in terms of prediction performance. Via the reinforcement learning mechanism, the fuzzy rule-based state space modeling and the adaptive action selection policy, the proposed system leads to higher directional predictability.

Acknowledgements

The author would like to thank the Editor Cars Hommes for his valuable suggestions. He also thanks Cees Diks and Ramazan Gençay for helpful discussions and comments on an earlier version of the paper. A part of the study was presented at the seminar series of the Center for Nonlinear Dynamics in Economics and Finance (CeNDEF) at the University of Amsterdam. The author acknowledges financial support from the Max Weber Programme at the European University Institute (EUI).

References

Adya, M., Collopy, F., 1998. How effective are neural networks at forecasting and prediction? A review and evaluation. Journal of Forecasting 17, 481–495.

Alexander, C., 1998. Volatility and correlation: measurement methods and applications. Risk Management and Analysis 1, 125–168.

Al-Shammari, M., Shaout, A., 1998. Fuzzy logic modeling for performance appraisal systems: a framework for empirical evaluation. Expert Systems with Applications 14, 238–323.

Altrock, C., 1997. Fuzzy Logic and Neurofuzzy Applications in Business and Finance. Prentice Hall, Upper Saddle River, New Jersey.

Antsaklis, P.J., Passino, K.M., 1993. An Introduction to Intelligent and Autonomous Control. Kluwer Academic Publishers, Norwell, MA.

Antunović, M., Cummer, S.A., 2004. Adaptive filter for event-based signal extraction. Automatika Journal 45, 129–135.

Barto, A.G., Sutton, R.S., Anderson, C.W., 1983. Neuron like elements that can solve difficult learning control problems. IEEE Transactions on Systems, Man, and Cybernetics 13, 835.

Bekaert, G., Wu, G., 2000. Asymmetric volatility and risk in equity markets. Review of Financial Studies 13, 1–42.

Berenji, H.R., 1991. Refinement of approximate reasoning-based controllers by reinforcement learning. In: Proceedings of the Eighth International Workshop Machine Learning, pp. 475–479.

Berenji, H.R., 1996. Fuzzy Q-learning for generalization of reinforcement learning. In: Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, New Orleans, Louisiana, pp. 2208–2214.

Bertsekas, D.P., Tsitsiklis, J.N., 1996. Neuro-dynamic Programming. Athena Scientific.

Black, F., 1986. Noise. Journal of Finance 41, 529–543.

Bolger, F., Harvey, N., 1995. Judging the probability that the next point in an observed time-series will be below, or above, a given value. Journal of Forecasting 14, 597–607.

Bollerslev, T., 1986. Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics 31, 307–327.

Brock, W., Hsieh, D., LeBaron, B., 1991. Nonlinear Dynamics, Chaos and Instability. The MIT Press.

Brock, W.A., Hommes, C.H., 1998. Heterogeneous beliefs and routes to chaos in a simple asset pricing model. Journal of Economic Dynamics and Control 22, 1235–1274.

Buckley, J.J., Hayashi, Y., 1994. Fuzzy neural networks: a survey. Fuzzy Sets and Systems 66, 1–13.

Camillo, L., 2008. A combined signal approach to technical analysis on the S&P 500. Journal of Business and Economics Research 6, 41–51.

Christie, A., 1982. The stochastic behavior of common stock variances-value, leverage and interest rate effects. Journal of Financial Economics 10, 407–432.

Christoffersen, P., Diebold, F., 2006. Financial asset returns, direction-of-change forecasting, and volatility dynamics. Management Science 52, 1273–1287.

Cybenko, G., 1989. Approximation by superposition of a sigmoidal function. Mathematics of Control, Signals and Systems 2, 303–314.

De Lima, P.J.F., 1996. Nuisance parameter free properties of correlation integral based statistics. Econometric Reviews 15, 237–259.

Elman, J.L., 1990. Finding structure in time. Cognitive Science 14, 179–211.

Er, M.J., Deng, C., 2004. Online tuning of fuzzy inference systems using dynamic fuzzy Q-learning. IEEE Transactions on Systems, Man and Cybernetics—Part B: Cybernetics 34, 1478–1489.

Ersoy, O., 1990. In: Tutorial at Hawaii International Conference on Systems Sciences.

Fama, E.F., 1970. Efficient capital markets: a review of empirical work. Journal of Finance 25, 383–417.

Fama, E.F., 1991. Efficient capital markets II. Journal of Finance 46, 1575–1617.

Fama, E.F., French, K.R., 1995. Size and book-to-market factors in earnings and returns. Journal of Finance 50, 131–155.

Fama, E.F., Blume, M.E., 1966. Filter rules and stock-market trading. Journal of Business 39, 226–241.

Fernández-Rodriguez, F., Gonzalez-Martel, C., Sosvilla-Rivero, S., 2000. On the profitability of technical trading rules based on artificial neural networks: evidence from the Madrid stock market. Economics Letters 69, 89–94.

Franses, P.H., van Dijk, D., 2000. Non-linear Time Series Models in Empirical Finance. Cambridge University Press, Cambridge.

Funahashi, K., 1989. On the approximate realization of continuous mappings by neural networks. Neural Networks 2, 183–192.

Gallant, A.R., White, H., 1988. There exists a neural network that does not make avoidable mistakes. In: Proceedings of the Second Annual IEEE Conference on Neural Networks, San Diego, CA. IEEE Press, New York, vol. I, pp. 657–664.

Gallant, A.R., White, H., 1992. On learning the derivatives of an unknown mapping with multiplayer feedforward networks. Neural Networks 5, 129–138.

Gençay, R., 1998a. The predictability of security returns with simple technical trading rules. Journal of Empirical Finance 5, 347–359.

Gençay, R., 1998b. Optimization of technical trading strategies and the profitability in security markets. Economics Letters 59, 249–254.

Giot, P., 2005. Relationships between implied volatility indexes and stock index returns. Journal of Portfolio Management, 92–100.

Glorennec, P.Y., 1993. Fuzzy Q-learning and dynamic fuzzy Q-learning. In: Proceedings of the Third IEEE International Conference on Fuzzy Systems, Orlando, vol. 1, pp. 474–479.

Glorennec, P.Y., 2000. Reinforcement learning: an overview. In: European Symposium on Intelligent Techniques, Aachen, Germany, mimeo.

Glorennec, P.Y., Jouffe, L., 1996. A reinforcement learning method for an autonomous robot. In: Proceedings of the EUFIT'96, Fourth European Congress in Intelligent Technology, Soft Computing, Aachen, Germany, pp. 1100–1104.

Glorennec, P.Y., Jouffe, L., 1997. Fuzzy Q-learning. In: Proceedings of the Sixth IEEE International Conference on Fuzzy Systems, Barcelona, Spain, pp. 659–662.

Glosten, L., Jaganathan, R., Runkle, D., 1993. On the relation between the expected value and the volatility of the nominal excess return on stocks. Journal of Finance 48, 1779–1801.

Golub, G.H., Reinsch, C., 1971. Singular Value Decomposition and Least Squares Solutions: Handbook for Automatic Computation, vol. II: Linear Algebra. Springer, Heidelberg.

Golub, G.H., van Loan, C.F., 1989. Matrix Computations, 2nd Eds The Johns Hopkins University Press.

Gullapalli, V., 1992. Reinforcement learning and its application to control. Ph.D. Thesis, Department of Computer and Information Sciences, University of Massachusetts, Amherst.

Gradojevic, N., 2007. Non-linear, hybrid exchange rate modelling and trading profitability in the foreign exchange market. Journal of Economic Dynamics and Control 31, 557–574.

Green, H., Pearson, M., 1994. Neural nets for foreign exchange trading. In: Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets, Wiley, New York.

Hamilton, J.D., 1989. A new approach to the economic analysis of nonstationary time series subject to changes in regime. Econometrica 57, 357–384.

Hamilton, J.D., 1994. Time Series Analysis. Princeton University Press, Princeton.

Hecht-Nielsen, R., 1989. Theory of the backpropagation neural networks. In: Proceedings of the International Joint Conference on Neural Networks, Washington, DC, IEEE Press, New York, vol. I, pp. 593–605.

Henriksson, R.D., Merton, R.C., 1981. On the market timing and investment performance II: statistical procedures for evaluating forecasting skills. Journal of Business 54, 513–533.

Hommes, C.H., 2001. Financial markets as complex adaptive evolutionary systems. Quantitative Finance 1, 149–167.

Hommes, C.H., 2006. Heterogeneous agent models in economics and finance. In: Tesfatsion, L., Judd, K.L. (Eds.), Handbook of Computational Economics, vol. 2: Agent-Based Computational Economics. Elsevier Science B.V, pp. 1109–1186.

Horn, R.A., Johnson, C.R., 1991. Topics in Matrix Analysis. Cambridge University Press.

Hornik, K., 1991. Approximation capabilities of multilayer feedforward net-works. Neural Networks 4, 251–257.

Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural Networks 2, 359–366.

Hornik, K., Stinchcombe, M., White, H., 1990. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. Neural Networks 3, 551–560.

Hsu, P.-H., Kuan, C.-M., 2005. Reexamining the profitability of technical analysis with data snooping checks. Journal of Financial Econometrics 3, 606–628.

Hull, J.C., 2000. Options, Futures, and Other Derivatives, 4th ed Prentice Hall, Upper Saddle River, NJ.

Irwin, S.H., Park, C.H., 2007. What do we know about the profitability of technical analysis? Journal of Economic Surveys 21, 786–826

Ishibuchi, H., Kwon, K., Tanaka, H., 1995. A learning algorithm of fuzzy neural networks with triangular fuzzy weights. Fuzzy Sets and Systems 71, 277–293.

Jamshidi, M., Titli, A., Zadeh, L., Boverie, S., 1997. Applications of Fuzzy Logic. Prentice Hall, New Jersey.

Jasic, T., Wood, D., 2004. The profitability of daily stock market indices trades based on neural network predictions: case study for the S&P 500, the DAX, the TOPIX, and the FTSE in the period 1965–1999. Applied Financial Economics 14, 285–297.

Jorion, P., 2000. Value at Risk, 2nd eds McGraw-Hill, New York.

Jouffe, L., 1998. Fuzzy inference systems learning by reinforcement methods. IEEE Transactions on Systems, Man and Cybernetics—Part C, Applications and Reviews 28, 338–355.

Kaelbling, L.P., Littman, L.M., Moore, A.W., 1996. Reinforcement learning: a survey. Journal of Artificial Intelligence Research 4, 237–285.

Kao, G.W., Ma, C.K., 1992. Memories, heteroscedasticity and prices limit in currency futures markets. Journal of Futures Markets 12, 672–692.

Katz, J.O., 1992. Developing neural network forecasters for trading. Technical Analysis of Stocks and Commodities 58–70.

Kaufman, J.P., 1998. Trading Systems and Methods, 3rd eds John Wiley & Sons, New York.

Kimura, H., Kobayashi, S., 1998. An analysis of actor/critic algorithms using eligibility traces: reinforcement learning with imperfect value functions. In: Proceedings of the ICML-98, pp. 278–286.

Kirkpatrick, C.D., Dahlquist, J.R., 2007. Technical Analysis: The Complete Resource for Financial Market Technicians. Financial Times Press, Upper Saddle River, New Jersey.

Klir, G.J., Yuan, B., 1995. Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall, Upper Saddle River.

Kosko, B., 1992. Neural Networks and Fuzzy Systems. Prentice-Hall, Englewood Cliffs.

Krugman, P., 1987. Trigger strategies and price dynamics in equity and foreign exchange markets. NBER Working Paper No. 2459.

Kuan, C.-M., White, H., 1994. Artificial neural networks: an econometric perspective. Econometric Reviews 13, 1–91.

La Porta, R., Lakonishok, J., Shliefer, A., Vishny, R., 1997. Good news for value stocks: further evidence on market efficiency. Journal of Finance 52, 859–874.

Lawrence, M., O'Connor, M., 1992. Exploring judgemental forecasting. International Journal of Forecasting 8, 15–26.

Levich, R.M., Thomas, L.R., 1993. The significance of technical trading rule profits in the foreign exchange market: a bootstrap approach. In: Strategic Currency Investing-Trading and Hedging in the Foreign Exchange Market, Probus, Chicago, pp. 336–365.

Lin, C.-T., Lee, C.G., 1996. Neural Fuzzy Systems. A Neuro-fuzzy Synergism to Intelligent Systems. Prentice Hall, New York.

Lo, A.W., Mamaysky, H., Wang, J., 2000. Foundations of technical analysis: computational algorithms, statistical inference, and empirical implementation. Journal of Finance 55, 1705–1765.

Mamdani, E., 1977. Application of fuzzy logic to approximate reasoning using linguistic systems. IEEE Transactions on Computers 26, 1182–1191.

Manger, R., 1994. Using holographic neural networks for currency exchange rates prediction. In: Proceedings of the 16th International Conference on Information Technology Interface, Pula, Croatia.

Masters, T., 1993. Practical Neural Network Recipes in C++. Morgan Kaufmann, Academic Press.

Murphy, J.J., 1999. Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications. Prentice Hall Press, Paramus, New Jersey.

Nishina, T., Hagiwara, M., 1997. Fuzzy inference neural network. Neurocomputing 14, 223–239.

Plummer, T., Ridley, A., 2003. Forecasting Financial Markets: The Psychology of Successful Investing, London, Kogan.

Poddig, A., 1993. Short term forecasting of the USD/DM exchange rate. In: Proceedings of the First International Workshop on Neural Networks in Capital Markets, London.

Rawani, A.M., Mohapatra, D.K., Srinivasan, S., Mohapatr, P.K.J., Mehta, M.S., Rao, G.P., 1993. Forecasting and trading strategy for the foreign exchange market. Information and Decision Technologies 19, 55–62.

Schwert, W., 2002. Stock volatility in the new millennium: how wacky is Nasdaq? Journal of Monetary Economics 49, 3–26

Shiller, R.J., 1989. Investor behavior in the October 1987 stock market crash: survey evidence. NBER Working Paper No. 2446.

Shiller, R.J., 2002. From efficient market theory to behavioral finance. Cowles Foundation Discussion Paper No. 1385.

Shleifer, A., Summers, L.H., 1990. The noise trader approach to finance. Journal of Economic Perspectives 4, 19–33.

Simon, D., 2003. The Nasdaq volatility index during and after the bubble. Journal of Derivatives, 9–23.

Simon, H.A., 1957. Models of Man. Wiley, New York, NY.

Sugeno, M., 1985. Industrial Applications of Fuzzy Control. Elsevier Science Publications.

Sugeno, M., 1988. Fuzzy Control. Nikkan Kougyou, Shinbunsha.

Sutton, R.S., 1989. Learning to predict by the method of temporal differences. Machine Learning 3, 9–44.

Sutton, R.S., Barto, A.G., 1998. Reinforcement Learning: An Introduction. The MIT Press, Cambridge.

Sutton, R.S., McAllester, D., Singh, S., Mansour, Y., 2000. Policy gradient methods for reinforcement learning with function approximation. Advances in Neural Information Processing Systems 12, 1057–1063.

Tay, N.S.P., Linn, S.C., 2001. Fuzzy inductive reasoning, expectation formation and the behavior of security prices. Journal of Economic Dynamics and Control 25, 321–361.

Watkins, C., 1989. Learning from delayed rewards. Ph.D. Thesis, University of Cambridge, England.

Weigend, A.S., 1991. Generalization by weight-elimination applied to currency exchange rate prediction. In: Proceedings of the IEEE International Joint Conference on Neural Networks, Singapore.

Whaley, R., 2000. The investor fear gauge. Journal of Portfolio Management 26, 12–27.

White, H., 1989. Learning in artificial neural networks: a statistical perspective. Neural Computing 1, 425–464.

Yao, J.T., Poh, H.-L., Jasic, T., 1996. Foreign exchange rates forecasting with neural networks. In: Proceedings of the International Conference on Neural Information Processing, Hong Kong, pp. 754–759.

Zhang, X.R., 1994. Non-linear predictive models for intra-day foreign exchange trading. International Journal of Intelligent Systems in Accounting, Finance and Management 3, 293–302.

Zimmerman, H.-J., Thole, U., 1978. On the suitability of minimum and product operators for the intersection of fuzzy sets. Fuzzy Sets and Systems 2, 173–186.