Comput Manag Sci (2012) 9:89–107
DOI 10.1007/s10287-011-0131-1

ORIGINAL PAPER

Regime-switching recurrent reinforcement learning for investment decision making

Dietmar Maringer · Tikesh Ramtohul

Received: 19 December 2009 / Accepted: 3 August 2011 / Published online: 10 September 2011

Abstract This paper presents the regime-switching recurrent reinforcement learning (RSRRL) model and describes its application to investment problems. The RSRRL is a regime-switching extension of the recurrent reinforcement learning (RRL) algorithm. The basic RRL model was proposed by Moody and Wu (Proceedings of the IEEE/IAFE 1997 on Computational Intelligence for Financial Engineering (CIFEr). IEEE, New York, pp 300–307 1997) and presented as a methodology to solve stochastic control problems in finance. We argue that the RRL is unable to capture all the intricacies of financial time series, and propose the RSRRL as a more suitable algorithm for such type of data. This paper gives a description of two variants of the RSRRL, namely a threshold version and a smooth transition version, and compares their performance to the basic RRL model in automated trading and portfolio management applications. We use volatility as an indicator/transition variable for switching between regimes. The out-of-sample results are generally in favour of the RSRRL models, thereby supporting the regime-switching approach, but some doubts exist regarding the robustness of the proposed models, especially in the presence of transaction costs.

1 Introduction

The recurrent reinforcement learning (RRL), proposed by Moody and Wu (1997), is a direct reinforcement approach for investment decision making. It has an autoregressive outlook and can be likened to a recurrent neural network with a single layer. Previous work has already shown that the RRL offers good promise in finding

D. Maringer (✉) · T. Ramtohul
Universität Basel, 4002 Basel, Switzerland
e-mail: Dietmar.Maringer@unibas.ch

T. Ramtohul
e-mail: Tikesh.Ramtohul@unibas.ch

profitable strategies in financial markets. Despite the reported findings, its simplistic nature casts some doubts about its ability to capture the non-linearities present in financial data. This is the motivating factor behind our study. We propose a new model, called regime-switching recurrent reinforcement learning (RSRRL), that augments the existing RRL with regime-switching properties to cater for these non-linearities. The principal goal of this paper is to give a detailed description of this new model, and compare its performance with the basic RRL in investment applications. We look at two variants of the RSRRL, a threshold version (TRRL) and a smooth transition (STRRL) version. We perform controlled experiments using artificial data to better understand the working principles of both sets of algorithms, and then use real-world data sets to test the efficiency of the systems in a simple automated trading setting. Additionally, an active portfolio management strategy based on these models is investigated and the performance of the three types of investors is compared with a passive benchmark.

The outline of the paper is as follows: in Sect. 2, we present a review of previous work concerned with the application of RRL in financial trading. Section 3 is devoted to the RSRRL. It starts by briefly reviewing the RRL methodology and proceeds with a detailed description of the RSRRL model, with emphasis on the learning procedure and the selection of indicator/transition variables. The ensuing section describes the experiments carried out to compare the two methodologies, presents the results, and provides an assessment of the main findings. Section 5 provides the concluding remarks and discusses possibilities for future work.

2 Literature review

Early work by Moody and Wu (1997) and Moody et al. (1998) aimed at demonstrating the efficiency of the RRL methodology for training trading systems and portfolios by optimising the differential Sharpe ratio (DSR). Their early studies emphasized on two main aspects. First, trading systems based on the reinforcement learning paradigm perform better than those based on supervised learning techniques. Second, mechanical traders trained to maximise a risk-adjusted performance criterion like the DSR outperform trading systems which aim at either maximising profits or minimising some error criterion. Based on these results, Moody and Saffell (2001) used real data sets to test the efficacy of the RRL-traders. They used the half-hourly US Dollar/British Pound FX rate from the first 8 months of quotes in 1996 to train a 3-position, i.e. $\{long, short, neutral\}$ trader. The differential downside deviation ratio (see Moody and Saffell 2001) was used as the performance criterion. The RRL-traders led to profitable situations and positive Sharpe ratios in both the absence and presence of transaction costs, thereby indicating the ability of the RRL technique to discover structure in real-world financial data series. The authors also compared the performance of the RRL-trader with a Q-trader (a reinforcement learning approach developed by Watkins (1989)) and a simple buy-and-hold strategy for an asset allocation problem between the S&P 500 and T-bills for a 25-year period (1970–1994). It was found that both sets of RRL-traders and Q-traders yield higher Sharpe ratios than the buy-and-hold strategy, suggesting that reinforcement learning approaches are able to uncover useful

patterns. And interestingly, RRL-traders outperformed the Q-traders in all aspects, be it performance, interpretability or computational efficiency, thereby enhancing the appeal for direct reinforcement learning approaches in the design of trading systems.

As a follow-up work on the single-layer RRL technique, Gold (2003) extended the model to a two-layer neural network and subsequently drew comparisons between the effectiveness of this variant with the original single-layer network. He used half-hourly quotes from 25 different FX markets for the entire year of 1996. The traders were of the $\{long, short\}$ type and the DSR was the objective function used for optimising the network weights. The author also performed some tuning to obtain good candidate values for some key model parameters like the learning rate and number of training epochs. His results showed that the RRL-traders were profitable in most markets, although for a select few, very low and even negative Sharpe ratios were reported. Despite the slightly mitigated performance, the general impression was that the RRL algorithm is able to capture certain patterns and come up with profitable situations. Moreover, the results also demonstrated that better performance is obtained with the one-layer network than with the two-layer version. The author attributed this to noisy financial data. He claimed that the more intricate version overfits the data and tentatively pointed out that trading in FX markets might not require models that are too complex.

A full-fledged automated trading system based on the RRL was put forward by Dempster and Leemans (2006). They used a slightly modified version of the basic RRL as part of a trading system with a layered structure for trading in FX markets. The system consists of a machine learning layer, a risk management layer and a dynamic utility optimisation layer. The purpose of the risk management layer is to subject the output of the machine learning layer to certain risk constraints before the final trading decision is taken. The main role of the dynamic optimisation layer is to find optimal values for the model parameters in an adaptive fashion. They used one-minute data for the Euro-Dollar currency pair, spanning a period from January 2000 upto January 2002. The results showed that the risk management layer and the dynamic utility optimisation layer give rise to better performance, hence implying that such a layered structure might be worth considering while designing fully automated trading systems. An important point reported by the authors concerns the use of inputs other than lagged returns to the RRL. They experimented with various popular technical indicators as input, but did not notice any added improvement in performance. This led them to conclude that the RRL algorithm is able to efficiently exploit structure in past returns time series.

More recently, Bertoluzzo and Corazza (2007) used the RRL algorithm to develop a $\{long, short, neutral\}$ trading system, and applied it to nine of the major world financial market indices for the period between April 1992 and March 2007. The model is similar to the one proposed by Moody and Wu (1997) except that the authors used the reciprocal of the returns weighted direction symmetry index[1] as their maximisation criterion instead of the DSR. Daily closing prices were considered instead of high-frequency data. Moreover, a stop-loss criterion was included to prevent large

[1] It corresponds to the ratio of the cumulative positive trading returns to the cumulative negative trading returns.

drawdowns. Once more, results were very encouraging; the RRL-traders led to profitable situations in all but one case.

3 Model description

3.1 Recurrent reinforcement learning

Reinforcement Learning (RL) is a type of machine learning technique which focuses on goal-directed learning from interaction (Sutton and Barto 1998). It is a way of programming agents by reward and punishment without needing to specify how the task is to be achieved (Kaelbling et al. 1996); in other words, the learning process does not require target outputs, and is therefore different from supervised learning which is based on the availability of input/output pairs for training. RL can be used to find approximate solutions to stochastic dynamic programming problems and it can do so in an online fashion (Moody et al. 1998). In the last decade or so, it has attracted rapidly growing interest in the computational finance community, especially for the design of trading systems. The RRL, proposed by Moody and Wu (1997), is one such algorithm that uses the reinforcement paradigm to make investment decisions. It is an adaptive policy search algorithm which tries to maximise a certain performance criterion in order to learn profitable investment strategies. As its name suggests, the system is recurrent, meaning that the current investment decision has a say in shaping future decisions. In the presence of transaction costs, investment performance depends on sequences of interdependent decisions; the recurrent nature of the algorithm takes this path-dependency into account (Moody et al. 1998). Moody and Saffell (2001) describe the RRL as a computationally efficient algorithm that allows for simpler problem representation, avoids Bellman's curse of dimensionality, and circumvents problems that are generally associated with trading systems based on price forecasts.

The RRL model can be thought of as a gradient ascent algorithm which aims at optimising some desired criterion. The basic version was developed to trade fixed position sizes in a single security, but it can easily be extended to trade in varying quantities, or to manage multiple asset portfolios (see Moody et al. 1998), or for asset allocation (see Moody et al. 1998; Moody and Saffell 2001). A single-asset, two-position trader will be discussed in this paper. The trader can take only long or short positions of constant magnitude. Neutral positions are not allowed, so he is always in the market; this is also known as a reversal system (Gold 2003). The trading function is as follows:

$$F_t = \tanh\left(\sum_{i=0}^{m-1} w_i r_{t-i} + w_m F_{t-1} + w_{m+1} v\right). \tag{1}$$

F_t is the output of the network at time t. A long position is adopted when $F_t > 0$; the trader buys an asset at time t and makes a profit if the price goes up in the next time step. If $F_t < 0$, the trader short sells an asset at time t and makes a profit if the price goes down at time $t + 1$. If a three-position trader were to be considered, two cut-off points, f_s and f_b, need to be chosen such that $-1 < f_s < 0 < f_b < 1$; then, a long position is adopted when $F_t > f_b$, a short position when $F_t < f_s$, and

a neutral position when $f_s \leq F_t \leq f_b$. These thresholds can be set arbitrarily or some search/optimisation technique can be employed for finding the most appropriate values.

The price return r_t corresponds to the difference in value of the asset between the previous period and the current period, i.e. $r_t = p_t - p_{t-1}$. The term v is the familiar bias present in neural network models, typically having a value of 1. The w_i's denote the system parameters or network weights that need to be optimised. Note that the time indexation of the weights has been dropped for clarity. The term F_{t-1}, i.e. the trade position at the previous time step, induces recurrence and hence some kind of internal memory. The RRL model is not restricted to taking only lagged price returns as inputs. It can easily accommodate technical indicators or other economic variables that might have an impact on the security.

3.2 Regime-switching recurrent reinforcement learning

Despite the relative success of the single-layer RRL model, it can be argued that its linear outlook makes it ill-suited to capture all the intricate aspects of financial data. An approach with a higher degree of non-linearity could very much aid in increasing its predictive capabilities. One straightforward way of accounting for the non-linearities is to incorporate hidden layers in the network. But, Gold (2003) noted a decline in performance when he introduced a hidden layer in the RRL topology. Indeed, multilayer models are prone to overfitting, especially with noisy financial data, and are quite often unable to generalise properly. Moreover, such black-box approaches render inference about the input–output relationship difficult, if not impossible. A certain degree of transparency ensures that automated trading systems are more tractable, thereby allowing the human expert to adopt remedial measures or perform fine-tuning more efficiently whenever performance starts to degenerate. There is a need for non-linear models that can perform well out-of-sample and that can shed some light on how economic variables affect financial markets. Regime-switching models provide an elegant solution to this kind of problem. These models define different states of the world (regimes), and assume that the dynamic behaviour of economic variables depends on the regime that occurs at any given point in time. This implies that certain properties of the time series, such as its mean, variance, autocorrelation, etc., are different in different regimes. Such models offer a great deal of transparency and the concept of regimes helps to capture non-linearities. Moreover, the regime-switching framework is more adapted for modelling dramatic changes in behaviour in economic time series, as a consequence of events such as financial crises or major changes in government policy (Hamilton 2008).

There exists some well-established regime-switching methods that have gained prominence in econometrics. These include the threshold model, initially proposed by Tong (1978), the Markov-Switching model of Hamilton (1989), the artificial neural network model of White (1989), and the smooth transition model (see Teräsvirta 1994), the latter being a more general version of the threshold model. In this study, the threshold and smooth transition versions have been considered because of their simplicity and the degree of transparency that they offer. Suppose that we have a

2-regime situation for some dependent variable y_t, transition/indicator variable q_t and a threshold value c. Assuming that each regime is characterised by an $AR(1)$ process, the regime-switching model can be expressed as

$$\begin{aligned} y_t &= (\phi_{0,1} + \phi_{1,1} y_{t-1})(1 - G_t) + (\phi_{0,2} + \phi_{1,2} y_{t-1})(G_t) + z_t \\ G_t &= \begin{cases} I[q_t > c] & \text{for TAR} \\ [1 + \exp(-\gamma[q_t - c])]^{-1} & \text{for STAR} \end{cases} \end{aligned} \tag{2}$$

where z_t denotes an i.i.d white noise process, TAR stands for 'threshold autoregressive' and STAR denotes 'smooth transition autoregressive'. The values G_t for the threshold model are binary values while for the smooth transition version, G_t can take any value in the range [0 1]. The parameter γ dictates the smoothness of the transition. As γ tends to infinity, the logistic function approaches the indicator function. The interested reader is referred to Franses and van Dijk (2000) for more details about these models.

The RSRRL, viz the regime-switching version of the recurrent reinforcement learning algorithm, can be formulated by considering (1) and (2). To simplify the discussion, the focus will be on models that involve only two regimes. It is however trivial to extend the model to account for multiple regimes and/or multiple indicator variables. A two-regime system can be described as

$$\begin{aligned} F_t &= y_{t,1} G_t + y_{t,2}(1 - G_t) \\ y_{t,j} &= \tanh\left(\sum_{i=0}^{m-1} w_{i,j} r_{t-i} + w_{m,j} F_{t-1} + w_{m+1,j} v \right) \quad \text{for } j = \{1, 2\}. \end{aligned} \tag{3}$$

These systems can be thought of having two RRL networks (see Fig. 1), each one corresponding to a particular regime and having a distinct set of weights. The overall output F_t of the system is the weighted sum of the outputs $y_{t,1}$ and $y_{t,2}$ of the individual networks. The weighting factor is actually the value of the indicator/transition variable. Initially, both networks have the same set of weights. During training, the model promotes selective learning and this leads to each network developing a unique set of weights. If the system is in a particular regime, the network associated with that regime is exposed to higher weight updates than the other. For the threshold version (TRRL), each network learns a distinct mapping that corresponds to a specific region in the space spanned by the indicator variable. The latter effectively acts as a switch or gating device that selects the appropriate network at each time step. The smooth transition version (STRRL), on the other hand, allows a certain amount of overlap between the two regimes. The extent of the overlapping is regulated by the term γ.

3.3 Indicator/transition variable selection

There are many economic and financial variables that affect price movements in markets, but only a few actually can be regarded as potential candidates for switching between regimes in the RSRRL model. The very nature of the RRL calls for a transition

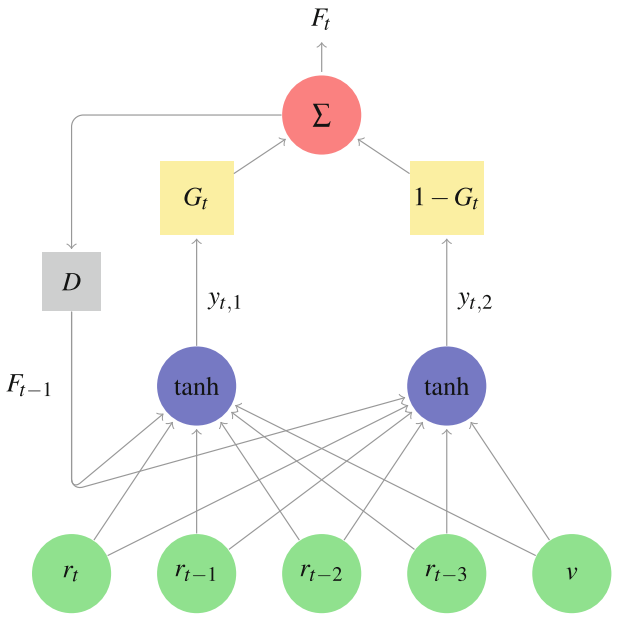

Fig. 1 RSRRL network structure where D is the delay operator and $m = 4$

variable that has certain desirable characteristics. First and foremost, it must have an impact on the serial correlation of the price returns process. This is a rather obvious requirement since the model takes lagged returns as inputs and is thus sensitive to the autocorrelations present in the data. Absence of any relationship between the indicator variable and serial correlation in the returns will most certainly lead to spurious learning. Next, for proper learning, the frequency of switching between regimes should be reasonable. Excessive switching tends to destabilise the learning process, while unreasonably low switching frequencies lead to situations where the system is reliant on very old information; this is not desirable since financial markets are dynamic, meaning that patterns that were present a decade ago might not be present now. On top of that, during the training phase, it is also important that the system goes through enough instances of each regime so that learning is not biased towards one network. Additionally, the indicator should preferably be an observable variable or a function of an observable variable that can readily and reliably be computed. Latent variables can therefore also be good candidates as long as the estimation process is fast and straightforward. The key strengths of the RRL are its speed and simplicity; an overly complex indicator might add too much of a computational burden and prove to be a deterring factor in the appeal of the algorithm. Last but not least, while choosing an indicator, one should bear in mind that the RRL inherently picks up trends in the data. Therefore, using trend or momentum indicators might add little value to the regime-switching model. Thus, the chosen indicator(s) should deliver some extra information about the market that the RRL cannot directly perceive. Also, if more than one indicator is to be used, the combination needs to be done in a smart way. Indicators should

deliver different type of information about the market and confirm each other rather than duplicate signals.

3.4 Differential Sharpe ratio for online learning

The learning process of the RSRRL is in essence similar to that of the RRL described in Moody and Wu (1997). It involves maximising a certain performance criterion to obtain a set of network weights that can lead to profitable strategies. Moody et al. (1998) showed that RRL systems trained by maximising risk-adjusted performance criteria perform better than those trained by minimizing error functions. They used stochastic gradient ascent to maximise the DSR, a variant of the well-known Sharpe ratio introduced by Sharpe (1966). The DSR is derived by making use of exponential moving average estimates of the first and second moments of the trading returns distribution. The same approach has been adopted in this paper. The trading return R_t, as defined by Moody et al. (1998), is expressed as

$$R_t = r_t^f + \text{sign}(F_{t-1})(r_t - r_t^f) - \delta|\text{sign}(F_t) - \text{sign}(F_{t-1})| \tag{4}$$

where r_t^f is the risk-free rate of interest and δ is the transaction cost rate per share traded. The exponential moving average Sharpe ratio can be expressed in terms of the trading return R_t. It is given by

$$S_t = \frac{A_t}{\sqrt{B_t - A_t^2}}, \tag{5}$$

where

$$A_t = A_{t-1} + \eta(R_t - A_{t-1}) = A_{t-1} + \eta\Delta A,$$
$$B_t = B_{t-1} + \eta(R_t^2 - B_{t-1}) = B_{t-1} + \eta\Delta B.$$

The DSR is obtained by expanding the exponential moving average version to first order in the adaptation rate η (Moody et al. 1998). It is given by

$$D_t = \frac{B_{t-1}\Delta A - \frac{1}{2}A_{t-1}\Delta B}{(B_{t-1} - A_{t-1}^2)^{\frac{3}{2}}}. \tag{6}$$

It can be optimised incrementally using gradient ascent. If ρ corresponds to the learning rate, the weight update equation is given by

$$w_{t,j} = w_{t-1,j} + \rho\Delta w_{t,j} \quad \text{for } j = \{1, 2\}, \tag{7}$$

where

$$\Delta w_{t,j} = \frac{dD_t}{dR_t}\left(\frac{dR_t}{dF_t}\frac{dF_t}{dw_{t,j}} + \frac{dR_t}{dF_{t-1}}\frac{dF_{t-1}}{dw_{t-1,\,j}}\right).$$

The derivative $\frac{dF_t}{dw_t}$ for online training can be computed using an approach similar to backpropagation through time (BPTT) introduced by Werbos (1990) and discussed in Moody et al. (1998),

$$\frac{dF_t}{dw_{t,j}} \approx \frac{\partial F_t}{\partial w_{t,j}} + \frac{\partial F_t}{\partial F_{t-1}} \frac{dF_{t-1}}{dw_{t-1,j}} \quad \text{for } j = \{1, 2\} \tag{8}$$

where

$$\frac{\partial F_t}{\partial w_{t,j}} = \frac{\partial F_t}{\partial y_{t,j}} \times \frac{\partial y_{t,j}}{\partial w_{t,j}},$$
$$\frac{\partial F_t}{\partial F_{t-1}} = \sum_{j=1}^{2} \left(\frac{\partial F_t}{\partial y_{t,j}} \times \frac{\partial y_{t,j}}{\partial F_{t-1}} \right).$$

All the required derivatives can be computed using basic differentiation rules, and thus the weight update process turns out to be rather straightforward and relatively fast.

4 Experiments

Three sets of experiments were carried out to gauge the performance of the basic RRL and the RSRRL models. The first one dealt with artificially generated data to illustrate the capabilities of the RSRRL to pick up trading signals in both one-regime and two-regime environments. The second set of experiments looked at how the different trading systems fared when faced with daily real financial data. The third experiment dealt with an active portfolio management strategy based on signals from the three models.

4.1 Methodology

The traders were of the {*long*, *short*} type and could only trade a fixed number (fraction) of shares at a time. If a trader is already in a certain position, he holds this position until the reverse trade signal is output by the system. The risk-free rate, r_t^f, in (4) has been assumed to be zero in all the experiments, which is reasonable since we are dealing with daily data[2]. The training phase consisted of allowing the traders to go through data of length L_{tr} for a number of epochs n_e. The performance of the traders was assessed by considering the trades made during an out-of-sample period L_{te}. The model parameters include the learning rate ρ, the adaptation rate η, the number of price return inputs m, the size of the training window L_{tr}, the number of training epochs n_e, and the size of the test window L_{te}. The values used were $L_{tr} = 2000$, $L_{te} = 375$, $m = 5$, $n_e = 5$, $\rho = 0.01$, and $\eta = 0.01$. These values are

[2] For a three-position trader, r_t^f cannot be overlooked despite its relatively low daily value since it might be beneficial for the trader to be out of the market for extended periods.

not optimal, but can be relied upon for proper learning. They are inspired from previous work by Moody and Saffell (2001) and Gold (2003) and based on preliminary results from simple grid search experiments that we carried out. For instance, 'good' candidates for n_e are integer values between three and six. Larger values for n_e tend to destabilise the learning process. Or, for the learning rate, relatively high values such as $\rho = 0.1$ corrupt the learning process. The initial weights were sampled from a uniform distribution such that $-0.1 \leq w_i \leq 0.1$. During learning, the weights are constrained within the range $-1.0 \leq w_i \leq 1.0$ to prevent saturation.

Because of the non-stationarity of the objective function, the optimisation process is not very stable. In particular, the models are very sensitive to the initial weights: traders having the same parameter settings except for the initial weights do not exhibit convergent behaviour during training. In this study, only the best performing traders during the in-sample period have been considered for out-of-sample trading. A bunch of traders were trained and their in-sample performance recorded. Only the top 1% (referred to as 'elitists' from hereafter) were considered for the test period. Note that this approach is probably not the most robust one, since in-sample performance does not always correlate positively with out-of-sample performance. In future, persistence in RRL/RSRRL behaviour could be investigated more thoroughly to determine the way forward while switching from the training period to the test period.

4.2 Artificial data series

A set of controlled experiments was conducted using artificial returns series to compare and contrast the behaviour of each bunch of traders. Two scenarios were considered, one in which the data series is characterised by a single regime and the second one consisting of data with regime-switching properties. The generic form of the data-generating process was as follows

$$
\begin{aligned}
r_t &= \begin{cases} \phi_0^{(1)} + \sum_{i=1}^{p} \phi_i^{(1)} r_{t-i} + a_t, & \text{if } \sqrt{h_t} > c \\ \phi_0^{(2)} + \sum_{i=1}^{p} \phi_i^{(2)} r_{t-i} + a_t, & \text{if } \sqrt{h_t} \leq c \end{cases} \\
a_t &= \sqrt{h_t} z_t, \quad z_t \sim N(0,1) \\
h_t &= \alpha_0 + \alpha_1 a_{t-1}^2 + \beta_1 h_{t-1}.
\end{aligned}
\tag{9}
$$

For the first scenario, both regimes were generated by AR processes with identical coefficients which effectively correspond to a single-regime situation. For the other scenario, the data series were constructed from the concatenation of two independent AR processes exhibiting autocorrelation of same magnitude but of different sign. In other words, the data from scenario 2 is made up of portions having either negative autocorrelation or positive autocorrelation. p was set to 2. For each scenario, 100 different realisations of the process were considered, and for each realisation, 10 'elitist' traders were picked for out-of-sample evaluation. Thus, an ensemble of 1,000 traders

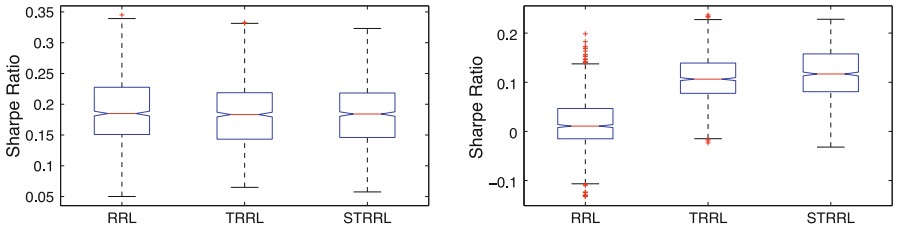

Fig. 2 Boxplots to compare the out-of-sample performance of an ensemble of 1,000 elitist traders for the different systems. The left panel corresponds to the single regime situation and the right panel is for the two-regime model. For the single-regime situation, the notches on the boxplots overlap, which imply that the median performance of the three systems is not significantly different from one another (5% level). For the two-regime scenario, the RSRRL systems perform significantly better than the RRL. First scenario: $\phi_0 = 0.01, \phi_1 = 0.2, \phi_2 = 0.1$. Second scenario: $\phi_0^{(1)} = -0.01, \phi_1^{(1)} = -0.2, \phi_2^{(1)} = -0.1$; $\phi_0^{(2)} = 0.01, \phi_1^{(2)} = 0.2, \phi_2^{(2)} = 0.1$

were considered for each model. The threshold c was set to 1, and the slope parameter γ for the STRRL was fixed to 20.

The results for this set of experiments are shown in Fig. 2. In the first scenario with a single regime, the standard RRL does not outperform the RSRRL models. It seems that both networks of the RSRRL systems are able to uncover the true data generating process. The corresponding weights in each branch typically have the same sign, although they might differ in magnitude. Thus, both branches tend to produce very similar trade signals after training, and the overall output of the RSRRL becomes almost regime-independent. Inspection of the results on a realisation-by-realisation basis showed that in some instances, the RSRRL outperform the RRL, while in others, the RRL does significantly better. It can be inferred that with data set with a single regime, or with regimes that are closely related to each other, the RSRRL models will on average match the performance of the RRL. The results for the second scenario indicate that the RSRRL systems perform significantly better than the RRL. This suggests that the standard RRL cannot deal with data sets in which the serial correlation changes sign from one portion to another. It seems that such data sets have a nullifying effect on the learning process. A network with a single set of weights cannot perform well in those two distinctly different regimes. Whenever there is a regime shift, the RRL takes time to adjust to its new environment and is unable to come up with profitable strategies. The RSRRL models, on the other hand, are able to avoid this pitfall, since they develop a specific set of weights for each regime.

4.3 Financial data series

The data consisted of daily closing prices of 12 randomly-chosen components from the Dow Jones Industrial Average (DJIA) index, namely ExxonMobil (XOM), Chevron Corporation (CVX), Johnson & Jonhnson (JNJ), Pfizer (PFE), Bank of America (BAC), United Technologies Corporation (UTX), Travelers (TRV), Wal-Mart (WMA), 3M (MMM), Procter & Gamble (PG), Dupont (DD) and Verizon Communications (VZ). A nine-year period from April 2000 upto September 2009 was considered.

Table 1 Ljung–Box hypothesis test results

	BAC	CVX	DD	JNJ	MMM	PFE	PG	TRV	UTX	WMT	VZ	XOM
L_{tr}	0.270	0.161	0.312	0.009	0.428	0.000	0.107	0.568	0.000	0.016	0.215	0.006
L_{te}	0.273	0.000	0.057	0.000	0.035	0.008	0.007	0.000	0.002	0.000	0.010	0.000

The p values for the Ljung–Box hypothesis test with null hypothesis of zero serial correlation. For the training period, the null hypothesis cannot be rejected at the 5% level for the majority of the data sets. The test period, however, shows strong evidence for serial correlation in nearly all samples

It was divided into a training portion consisting of 2,000 datapoints (roughly 8 years of data), and a test portion of 375 datapoints corresponding to the period between April 2008 and September 2009. From an economic viewpoint, the data sets reflect chronologically the end of the dot-com bubble, followed by the start of the US housing bubble and its subsequent deflation that culminated into the current crisis.

The stocks come from various sectors and their log return series exhibit varying amounts of autocorrelation. A Ljung–Box test was carried out to quantify the serial correlation for both the training and test periods. The p values are reported in Table 1. Volatility was used as indicator/transition variable for the RSRRL models. The motivation behind this choice stems from empirical studies carried by LeBaron (1992); Sentana and Wadhwani (1992); Koutmos (1997) and McKenzie and Faff (2003) which point towards a relationship between volatility and serial correlation. The authors found that a rise in the volatility level tends to increase the likelihood of negative autocorrelation in price returns. Moreover, because of its persistent nature and the well-known phenomenon of volatility clustering, there is little risk of the trading system suffering from excessive switching between regimes. If the switching frequency is too low, then a time-varying threshold can be used to address this issue. Despite its unobservability, various standard techniques that ally speed and reliability exist for the estimation of the volatility process.

The model used for the mean-volatility process is defined by (9) with $p = 5$. The parameter estimates were obtained by maximising the conditional likelihood function. Differential evolution (Storn and Price 1997) was used for the optimisation process. The datapoints from the training set were used for this purpose. In addition to the usual constraints associated with AR-GARCH modelling, it was ensured that each regime contains at least 40% of the observations. Because of the recursive nature of the GARCH process, the volatility forecasts for the out-of-sample period were readily available. The static threshold c was used to determine the values of the indicator/transition function for the RSRRL models. The slope parameter γ for the STRRL was set to 5. Figure 3 illustrates the relevant financial time series for the XOM data set, together with the threshold/transition values. The volatility profile shows how markets went from turbulent to tranquil, and then back to turbulent. The graphs for the indicator/transition function values depict this more clearly.

For each data set, the top 1% performers from an initial bunch of 1,000 were chosen for the evaluation phase. This was repeated 20 times, yielding a total of 200 'elitists' for each type of trading system. In each case, three types of scenarios were considered, one without transaction costs, one with $\delta = 5$ bp, and one with $\delta = 10$ bp (see (4)).

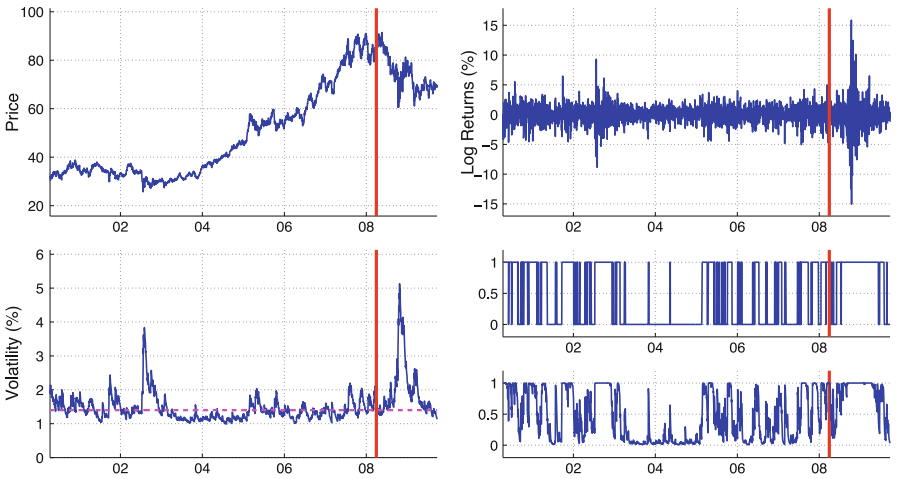

Fig. 3 The price series, return series, volatility estimates and indicator/transition values for the XOM data set (from April 2000 to September 2009). The delimiters separate the training set from the test set. The horizontal dotted lines in the graph for volatility correspond to the threshold used for the identification of regimes. The graphs on the bottom right depict how the indicator values for the TRRL (*upper graph*) and the transition values for the STRRL (*lower graph*) evolve. It can be seen that the test period is predominantly in a high-volatility regime

Tables 2, 3 and 4 summarise the out-of-sample results. In Table 2, the performance of the trading systems in the absence of transaction costs is presented. Table 4 gives the mean out-of-sample Sharpe ratios for all the scenarios, while Table 3 shows the influence of transaction costs on the trading frequency. The results without transaction costs are discussed first.

There are two main observations emanating from Table 2. First, all three models yield mostly positive Sharpe ratios. Next, the RSRRL models outperform the RRL in the majority of cases. The first point can be explained by looking at the Ljung–Box test results in Table 1. For almost all data sets, the out-of-sample period is marked by significant amounts of serial correlation, as a result of the financial crisis. Since the RRL methodology thrive on autocorrelation in the returns series, the trading systems are able to pick up the strong trends present in the data. It is not a coincidence that all three systems perform rather poorly on the BAC data set: the p values for both the training and test portions point towards insignificant serial correlation, and consequently the trading systems are not capable of finding good strategies. However, there is no clear-cut relationship between the amount of autocorrelation and the performance of the RRL systems. Since real-world financial time series are typically extremely noisy, it is almost impossible to determine an exact relationship. But, it seems that, despite the relatively high noise level, the RRL systems are able to spot the strong market sentiments and translate this into profitable situations.

The second point, relating to the RSRRL outperforming the RRL for the majority of the data sets studied, can be explained by considering the switch from a low-volatility to a high-volatility regime in the whereabouts of the start of the out-of-sample period. The RRL is unable to adjust to its new environment quickly enough, and therefore

Table 2 Out-of-sample results for $\delta = 0$

Dataset	Trader	Mean	Std	Median	LQ	UQ	Max	Min
BAC	RRL	−0.0129	0.0153	−0.0164	−0.0216	−0.0070	0.0296	−0.0483
	TRRL	0.0142	0.0173	0.0195	0.0095	0.0261	0.0526	−0.0385
	STRRL	−0.0185	0.0139	−0.0207	−0.0269	−0.0125	0.0340	−0.0553
CVX	RRL	0.0214	0.0092	0.0229	0.0184	0.0273	0.0377	−0.0064
	TRRL	0.0191	0.0193	0.0148	0.0046	0.0358	0.0618	−0.0226
	STRRL	0.0294	0.0237	0.0292	0.0122	0.0493	0.0826	−0.0370
DD	RRL	0.0480	0.0168	0.0528	0.0468	0.0564	0.0685	−0.0145
	TRRL	0.0397	0.0106	0.0386	0.0332	0.0458	0.0675	−0.0154
	STRRL	0.0496	0.0155	0.0514	0.0370	0.0602	0.0797	0.0120
JNJ	RRL	0.0266	0.0098	0.0276	0.0198	0.0330	0.0510	−0.0042
	TRRL	0.0404	0.0111	0.0428	0.0368	0.0468	0.0584	−0.0182
	STRRL	0.0304	0.0195	0.0337	0.0188	0.0450	0.0666	−0.0234
MMM	RRL	−0.0146	0.0197	−0.0209	−0.0248	−0.0081	0.0801	−0.0338
	TRRL	0.0343	0.0256	0.0315	0.0195	0.0437	0.1124	−0.0227
	STRRL	0.0233	0.0105	0.0241	0.0161	0.0305	0.0516	−0.0083
PFE	RRL	−0.0270	0.0066	−0.0263	−0.0303	−0.0239	−0.0120	−0.0578
	TRRL	0.1508	0.0363	0.1490	0.1231	0.1860	0.2183	0.0677
	STRRL	0.0962	0.0545	0.0955	0.0442	0.1380	0.1947	0.0198
PG	RRL	0.0882	0.0138	0.0883	0.0770	0.0984	0.1193	0.0581
	TRRL	0.0816	0.0135	0.0760	0.0729	0.0933	0.1118	0.0536
	STRRL	0.0626	0.0113	0.0642	0.0579	0.0673	0.1276	0.0353
TRV	RRL	0.0508	0.0060	0.0511	0.0497	0.0511	0.0972	0.0445
	TRRL	0.1217	0.0087	0.1219	0.1147	0.1317	0.1346	0.1016
	STRRL	0.1063	0.0055	0.1061	0.1019	0.1120	0.1156	0.0933
UTX	RRL	0.1095	0.0039	0.1095	0.1086	0.1129	0.1169	0.0917
	TRRL	0.0943	0.0076	0.0948	0.0906	0.0987	0.1130	0.0585
	STRRL	0.0938	0.0069	0.0931	0.0899	0.0989	0.1097	0.0576
VZ	RRL	0.0702	0.0070	0.0723	0.0666	0.0732	0.0912	0.0483
	TRRL	0.1024	0.0104	0.1001	0.0973	0.1085	0.1355	0.0778
	STRRL	0.1133	0.0117	0.1167	0.1053	0.1212	0.1347	0.0820
WMT	RRL	0.0553	0.0109	0.0560	0.0518	0.0606	0.1138	0.0216
	TRRL	0.0512	0.0122	0.0499	0.0428	0.0624	0.0736	0.0219
	STRRL	0.0493	0.0118	0.0484	0.0423	0.0576	0.0826	0.0216
XOM	RRL	0.0665	0.0097	0.0691	0.0632	0.0726	0.0909	0.0181
	TRRL	0.1253	0.0299	0.1376	0.1010	0.1478	0.1821	0.0494
	STRRL	0.1011	0.0162	0.0984	0.0938	0.1024	0.1595	0.0333

performs poorly. The RSRRL models however, develop two sets of weights, one for each regime. The first set of weights is well-suited for highly volatile periods, while the second set is more suited for tranquil periods. The high volatility in the test period implies that the RSRRL models base their trade decisions on the first set of weights

Table 3 Trading frequency (in %) over $L_{tr} + L_{te}$

Trader	δ (bp)	BAC	CVX	DD	JNJ	MMM	PFE	PG	TRV	UTX	VZ	WMT	XOM
RRL	0	33.7	36.2	53.9	53.8	37.3	38.0	36.0	77.7	33.2	39.9	45.8	43.3
	5	25.0	23.6	25.7	15.3	18.6	32.9	28.2	21.2	21.1	31.6	17.8	23.0
	10	15.8	17.3	19.6	11.7	15.2	22.3	21.4	19.9	17.1	20.7	16.0	20.0
TRRL	0	48.3	46.0	45.9	46.0	39.9	36.8	38.2	53.1	43.9	44.5	44.1	31.8
	5	30.3	37.2	40.8	25.2	28.2	31.0	31.5	46.1	28.9	33.3	34.2	26.9
	10	22.5	29.5	36.4	17.9	16.9	25.1	19.2	25.4	20.3	30.5	26.8	24.5
STRRL	0	38.7	39.5	47.4	45.6	37.8	37.8	41.5	56.3	37.0	40.7	45.4	28.7
	5	28.8	33.8	40.2	27.2	29.6	30.1	24.6	48.1	31.8	33.6	34.0	27.2
	10	22.9	29.1	38.5	17.6	25.4	22.8	19.9	24.0	24.3	31.4	28.6	22.6

Table 4 Mean out-of-sample Sharpe ratios for all three scenarios

Trader	δ (bp)	BAC	CVX	DD	JNJ	MMM	PFE	PG	TRV	UTX	VZ	WMT	XOM
RRL	0	−0.013	0.021	0.048	0.027	−0.015	−0.027	0.088	0.051	0.110	0.070	0.055	0.067
	5	−0.024	0.004	0.015	0.000	0.016	−0.043	0.083	0.021	0.068	0.051	0.087	0.056
	10	−0.020	−0.014	−0.034	−0.012	0.012	−0.053	0.070	0.090	0.078	0.107	0.082	0.046
TRRL	0	0.014	0.019	0.040	0.040	0.034	0.151	0.082	0.122	0.094	0.102	0.051	0.125
	5	−0.005	0.004	0.042	0.053	0.028	0.109	0.052	0.113	0.036	0.094	0.055	0.092
	10	−0.017	−0.015	−0.008	−0.007	0.020	0.058	0.019	0.056	0.020	0.112	0.033	0.089
STRRL	0	−0.019	0.029	0.050	0.030	0.023	0.096	0.063	0.106	0.094	0.113	0.049	0.101
	5	−0.026	−0.013	0.049	0.000	0.013	0.107	0.046	0.089	0.056	0.119	0.038	0.068
	10	−0.021	−0.000	0.010	−0.002	0.007	0.066	0.046	0.055	0.023	0.123	0.013	0.095

rather than the second. Thus, the significant regime change during the test period does not have a detrimental effect on the out-of-sample performance. It can be inferred that there is a difference in the sign on the serial correlation present in each volatility regime for most of the real-world data sets. This is in line with the empirical findings of LeBaron (1992); Sentana and Wadhwani (1992); Koutmos (1997) and McKenzie and Faff (2003) discussed earlier. Analysis of the weights developed by the RSRRL traders tend to support this phenomenon; for most of the data sets studied, the set of weights developed for the high-volatility regime is typically more negative than that developed for the low-volatility regime. Of course, this is the broad picture. In a couple of cases, the RSRRL models cannot match the performance of the standard RRL. It could be that the high-volatility regime experienced during the financial crisis has very different characteristics from the one seen during the training phase. Thus, the weights learned during training are not suitable to guide the RSRRL-traders through this new environment. Or, because of the high-level of noise in financial time series, the systems are unable to uncover the correct input-output mapping from the sample, and, because of the additional complexity of the RSRRL models, they are more prone to overfitting than the standard RRL. Regarding the comparative performance of the TRRL and STRRL models, the results are in favour of the TRRL, although

the evidence is not concrete. It is possible that the drastic regime change during the out-of-sample period supports the TRRL since the latter undergoes stronger weight updates during learning and is therefore more apt at capturing the market sentiment during that portion of the test period with an unusually high volatility.

The inclusion of transaction costs has an impact on overall performance of the traders, as can be expected. For traders with the same initial weights and same network parameters, but different values of δ in (4), the training phase will lead to the development of different trading strategies for each of them. The values for the trading frequency over the combined training and test periods in Table 3 confirm this. As the transaction cost levels increase, the trading frequency decreases. The traders hold their 'current' positions for longer periods, in a bid to prevent the generation of excessive transaction costs. The values in Table 4 correspond to the mean out-of-sample Sharpe ratios achieved after accounting for the desired level of transaction costs. In general, performance levels decrease as the amount of transaction costs increases, leading to negative Sharpe ratios in quite a few situations. Although the systems come up with strategies that involve fewer trades in the presence of transaction costs, the decrease in trading frequency cannot fully counterbalance the detrimental effect of these costs in the trading returns. Interestingly, in certain cases, for instance with VZ, the traders achieve comparable or even higher Sharpe ratios in the presence of trading costs. This hints towards a serious overfitting issue for this data set when traders are trained without transaction costs. This is a good example of how inclusion of transaction costs in the model can help uncover more robust strategies. Another notable observation from Table 4 is that the RSRRL systems have a higher trading frequency than the standard RRL traders, for the corresponding levels of transaction costs. Thus, for higher transaction cost levels, the RSRRL systems have a higher likelihood of being adversely affected in their performance levels. This is explored further in the next section.

4.4 Portfolio management

We investigate the performance of an active portfolio management strategy based on the signals generated by each type of trading system. Each investor holds an equally-weighted portfolio consisting of the 12 stocks previously discussed. He has an initial endowment of \$12, with \$1 invested in each stock, meaning that he holds a certain number (fraction) of shares n_i for each stock. At each time step, the investor rebalances his portfolio based on the long/short signals generated by the RRL/TRRL/STRRL systems for each stock. He either buys or short sells n_i shares of the ith stock depending on the signal generated by the system for that stock. Consider the following example for illustrating the strategy. Suppose that the investor is long in all the assets at time t. Now suppose that the system for the BAC data set generates a *sell* signal while the systems for the other data sets all generate *buy* signals. The investor holds his current position for these stocks, but sell $2 \times n_{\text{BAC}}$ shares to go short in BAC. The rebalanced portfolio is now 'long' in 11 stocks and 'short' in 1 stock.

The same settings as described in Sects. 4.1 and 4.3 have been used for this set of experiment. The 'elitist' traders were used to implement the active investment strategy for the out-of-sample period. For comparative purposes, a 'sell and hold' (SnH)

Table 5 Out-of-sample terminal log returns for the portfolio-based trading systems

Trader	δ	Mean	Std	Median	LQ	UQ	Max	Min
RRL	0	0.2794	0.0249	0.2797	0.2607	0.2975	0.3360	0.2193
	5	0.1885	0.0400	0.1831	0.1550	0.2197	0.3086	0.0882
	10	0.2147	0.0491	0.2179	0.1803	0.2509	0.3142	0.0728
TRRL	0	0.4621	0.0290	0.4631	0.4430	0.4823	0.5361	0.3720
	5	0.3619	0.0356	0.3641	0.3352	0.3837	0.4452	0.2636
	10	0.2129	0.0471	0.2126	0.1815	0.2470	0.3142	0.0855
STRRL	0	0.4068	0.0328	0.4062	0.3833	0.4296	0.5067	0.3337
	5	0.2952	0.0271	0.2946	0.2768	0.3134	0.3752	0.2205
	10	0.2310	0.0398	0.2292	0.2073	0.2594	0.3220	0.1264
SnH				0.1554				

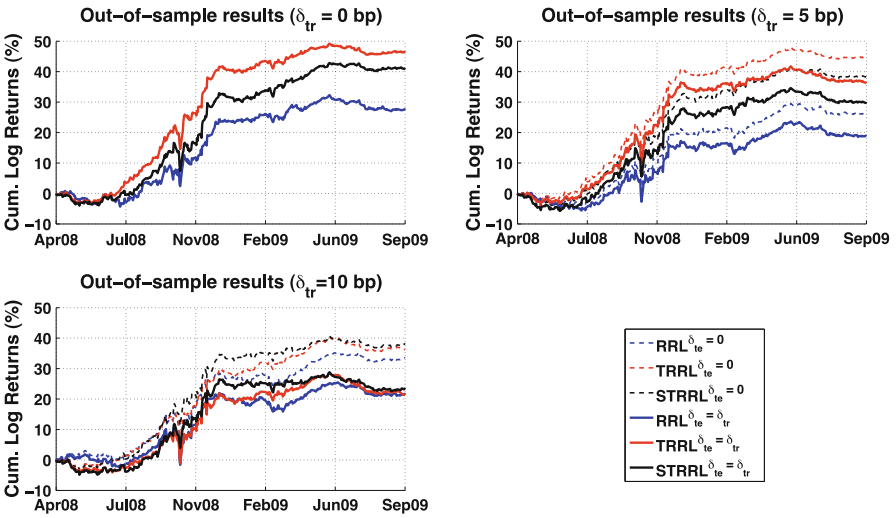

Fig. 4 Mean cumulative log returns of the equally-weighted portfolio based on trade recommendations from the three types of traders. The first graph corresponds to the scenario where $\delta = 0$, while the second and third graphs are for scenarios that included transaction costs during training. In these graphs, the dashed curves depict the mean performance of the traders in the absence of transaction costs during the test period. Note that the sell and hold strategy has a terminal log return of 15.5%

strategy was used as benchmark: the passive investor initially short sells n_i shares of each stock and does nothing during the investment horizon. The results are summarised in Table 5 in the form of the terminal log returns for each type of trader for different levels of transaction costs. It can be seen that for all cases studied, the active strategies outperform the passive benchmark, which again highlights the ability of the RRL/RSRRL systems to discover investment policies. Regarding the performance of the RSRRL systems relative to the standard RRL, the results are in favour of the former. Figure 4 depicts this in terms of the mean cumulative log returns over the out-of-sample period. For the scenarios involving trading costs, the dashed profiles correspond to

the mean performance of the trading strategies had these costs been overlooked during the test period. The discrepancy between the solid and dashed curves provide an insight into the influence of trading frequency and transaction costs on the behaviour of the traders over the different data series. It can be seen that the RSRRL systems are more seriously affected with an increase in δ, especially in the case where $\delta = 10$ bp. The point made in the last paragraph of Sect. 4.3 about the higher trading frequency of the RSRRL systems w.r.t. the standard RRL traders, is illustrated in the third graph. The discrepancy is larger for the RSRRL traders when transaction costs are ignored in the test phase, which imply that inclusion of these costs has a bigger incidence on their performance as a direct consequence of their higher trading frequency.

5 Conclusion

In this paper, we described the RSRRL model, which is an extended version of the RRL algorithm put forward by Moody and Wu (1997). We proposed two variants, namely a threshold model and a smooth transition version, and compared their performance with the basic RRL model in investment decision making. We used both artificial data and real-world data for our comparisons. We also emphasized on the importance of correct identification of the regimes, and advocated the use of volatility as a suitable indicator/transition variable.

Based on the simulation results with the artificial data, we found out that, in general, the performance of the RSRRL matches that of the RRL in data sets having a single regime. However, the RSRRL significantly outperform the RRL in situations where the data sets are characterised by distinctly different regimes. Results with the real data sets, in a simple automated trading framework, showed that the RSRRL models were superior in the majority of cases, and justified the use of volatility as the variable for defining the regimes. When applied to a portfolio management problem, it was found that active investment strategies based on signals from the RRL/RSRRL systems produced superior performance than a passive strategy. And once again, the RSRRL investors performed significantly better than the RRL investor. The results thus back the notion of integrating regime-switching with the RRL methodology, and also demonstrate the viability of using volatility as an indicator variable.

While results have been in favour of the RSRRL models, no general inference can be made about these models being consistently superior to the simple RRL system. For instance, with some data series, the RRL outperform their regime-switching variants. Thus, to get a more global idea, many more data series need to be considered, as well as multiple out-of-sample periods. Moreover, it was seen that the RSRRL trading systems exhibit a higher trading frequency than the RRL, which negatively impact their performance when transaction costs are considered. It is therefore important to investigate this area further by performing a more detailed sensitivity analysis to compare the robustness levels of these trading systems. Additionally, in the presence of trading costs, the 'neutral' position can be included in the trading systems, and the performance of these three-position traders relative to the two-position systems can be investigated.

For other market types of other data frequencies, volatility might not be the ideal indicator/transition variable. Different indicators might be required, either used in

conjunction with volatility, or completely independently. But since the RSRRL model is flexible, it can be easily modified to suit the financial problem or environment being investigated. It can be customised to match the needs of the problem and indicator variables combined in a variety of ways to best match the features of the application environment and the beliefs of the investor. It can easily be extended to accommodate more regimes and/or more indicator variables. If need be, the weighted average approach to compute the output can be altered, and more sophisticated techniques such as fuzzy inference can be implemented.

References

Bertoluzzo F, Corazza M (2007) Making financial trading by recurrent reinforcement learning. In: Knowledge-Based Intelligent Information and Engineering Systems and the XVII Italian Workshop on Neural Networks on Proceedings of the 11th International Conference. Springer-Verlag, USA, pp 619–626

Dempster M, Leemans V (2006) An automated FX trading system using adaptive reinforcement learning. Expert Syst Appl 30(3):543–552

Franses P, van Dijk D (2000) Nonlinear time series models in empirical finance. Cambridge University Press, Cambridge

Gold C (2003) FX trading via recurrent reinforcement learning. In: Proceedings. 2003 IEEE International Conference on Computational Intelligence for Financial Engineering, 2003. IEEE, pp 363–370

Hamilton JD (1989) A new approach to the economic analysis of nonstationary time series and the business cycle. Econometrica 57(2):357–384

Hamilton JD (2008) Regime-switching models. In: The New Palgrave Dictionary of Economics. Palgrave Macmillan, England

Kaelbling L, Littman M, Moore A (1996) Reinforcement learning: A survey. J Artif Intell Res 4(1):237–285

Koutmos G (1997) Feedback trading and the autocorrelation pattern of stock returns: further empirical evidence. J Int Money Financ 16(4):625–636

LeBaron B (1992) Some relations between volatility and serial correlations in stock market returns. J Bus 65(2):199–219

McKenzie MD, Faff RW (2003) The determinants of conditional autocorrelation in stock returns. J Financ Res 26(2):259–274

Moody J, Wu L (1997) Optimization of trading systems and portfolios. In: Proceedings of the IEEE/IAFE 1997 on Computational Intelligence for Financial Engineering (CIFEr). IEEE, New York, pp 300–307

Moody J, Wu L, Liao Y, Saffell M (1998) Performance functions and reinforcement learning for trading systems and portfolios. J Forecast 17(56):441–470

Moody J, Saffell M (2001) Learning to trade via direct reinforcement. IEEE Trans Neural Netw 12(4): 875–889

Sentana E, Wadhwani S (1992) Feedback traders and stock return autocorrelations: evidence from a century of daily data. Econ J 102(411):415–425

Sharpe W (1966) Mutual fund performance. J Bus 39(1):119–138

Storn R, Price K (1997) Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Glob optim 11(4):341–359

Sutton R, Barto A (1998) Introduction to reinforcement learning. MIT Press, Cambridge

Teräsvirta T (1994) Specification, estimation, and evaluation of smooth transition autoregressive models. J Am Stat Assoc 89(425):208–218

Tong H (1978) On a threshold model. In: Chen C (ed) Pattern recognition and signal processing. Sijthoff & Noordhoff, The Netherlands pp 101–141

Watkins C (1989) Learning from delayed rewards. Ph.D. thesis, University of Cambridge, England

Werbos P (1990) Backpropagation through time: what it does and how to do it. Proc IEEE 78(10):1550–1560

White H (1989) Some asymptotic results for learning in single hidden-layer feedforward network models. J Am Stat Assoc 84(408):1003–1013