Journal of Forecasting
J. Forecast. **17**, 441–470 (1998)

Performance Functions and Reinforcement Learning for Trading Systems and Portfolios

JOHN MOODY*, LIZHONG WU, YUANSONG LIAO AND MATTHEW SAFFELL

Oregon Graduate Institute of Science and Technology, USA

ABSTRACT

We propose to train trading systems and portfolios by optimizing objective functions that directly measure trading and investment performance. Rather than basing a trading system on forecasts or training via a supervised learning algorithm using labelled trading data, we train our systems using recurrent reinforcement learning (RRL) algorithms. The performance functions that we consider for reinforcement learning are profit or wealth, economic utility, the Sharpe ratio and our proposed *differential Sharpe ratio*. The trading and portfolio management systems require prior decisions as input in order to properly take into account the effects of transactions costs, market impact, and taxes. This temporal dependence on system state requires the use of reinforcement versions of standard recurrent learning algorithms. We present empirical results in controlled experiments that demonstrate the efficacy of some of our methods for optimizing trading systems and portfolios. For a long/short trader, we find that maximizing the differential Sharpe ratio yields more consistent results than maximizing profits, and that both methods outperform a trading system based on forecasts that minimize MSE. We find that portfolio traders trained to maximize the differential Sharpe ratio achieve better risk-adjusted returns than those trained to maximize profit. Finally, we provide simulation results for an S&P 500/TBill asset allocation system that demonstrate the presence of out-of-sample predictability in the monthly S&P 500 stock index for the 25 year period 1970 through 1994.

KEY WORDS trading systems; asset allocation; portfolio optimization; reinforcement learning; recurrent reinforcement learning; dynamic programming; on-line learning; recursive updating; Bernoulli utility; differential Sharpe ratio; transactions costs; state dependence; performance functions; recurrence

* Correspondence to: John Moody, Oregon Graduate Institute of Science and Technology, Department of Computer Science and Engineering, PO Box 91000, Portland, OR 97291-1000, USA. E-mail: moody@cse.ogi.edu

Contract grant sponsors: Nonlinear Prediction Systems and DARPA. Contract grant number: DAAH01-96-C-R026.

CCC 0277–6693/98/050441–30$17.50

INTRODUCTION: PERFORMANCE FUNCTIONS AND REINFORCEMENT LEARNING FOR TRADING

The investor's or trader's ultimate goal is to optimize some relevant measure of trading system performance, such as profit, economic utility or risk-adjusted return. In this paper, we propose to use reinforcement learning algorithms (on-line learning methods that find approximate solutions to stochastic dynamic programming problems) to directly optimize such trading system performance functions.* When transactions costs are included, the trading system must be recurrent, thus requiring a recurrent reinforcement learning (RRL) approach. This methodology can be applied to optimizing systems designed to trade a single security or to trade a portfolio of securities. For brevity, we refer to single security trading systems, asset allocation systems and multi-security portfolio management systems as *trading systems*.

The development of our approach is presented in the following three sections and simulation results are given in the fifth section. In the remainder of this section, we motivate our reinforcement learning approach by contrasting it with two conventional approaches to optimizing trading systems based on supervised learning. These include training a system to make price forecasts from which trading signals are generated and training on labelled trading data. Both supervised approaches are two-step procedures that are not guaranteed to globally optimize trading system performance.

Trading based on forecasts

A block diagram for a generic trading system based on forecasts is shown in Figure 1. In such a system, a forecast module is optimized to produce price forecasts from a set of input variables. Supervised learning techniques are used to minimize forecast error (typically mean squared error) on a training sample. The forecasts are then used as input to a trading module that makes buy and sell decisions or adjusts portfolio weights. Parameters of the trading module may be optimized, but are often set by hand.

Trading based on forecasts involves two steps, and minimizing forecast error is an *intermediate step* that is not the ultimate objective of the system. Moreover, the common practice of using only the forecasts as input to the trading module results in a loss of information relative to that available to the forecast module, in effect producing a *forecast bottleneck*. Both of these effects may lead to suboptimal performance.

Training a trading system on labelled data

It is possible to train a system to make trading decisions directly from the input variables, while avoiding the intermediate step of making forecasts. This is more direct, and avoids the forecast bottleneck. One technique for optimizing such a system is to use a supervised learning algorithm to train the system to make desired trades, as shown in Figure 2. A sequence of desired target trades (or portfolio weights) used for training the system is first determined via a *labelling procedure*. The data can be labelled by a human 'expert' or by an automatic labelling algorithm. The labelled trades are then used to train the trading system.

Training on labelled data is a two-step process. The procedure for labelling the data attempts to solve the *temporal credit assignment* problem, while subsequently training the system on the

* The terms *value function* (or *evaluation function*) and *objective function* are used in the reinforcement learning and optimization literatures, respectively. We prefer the term *performance function* for financial applications.

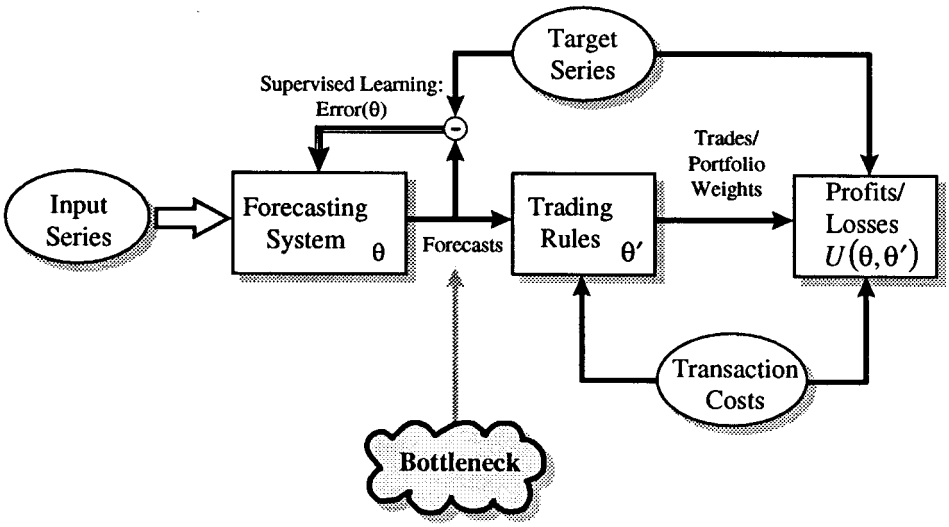

Figure 1. A trading system based on forecasts. The system includes a forecast module with adjustable parameters θ followed by a trading module with parameters θ'. Price forecasts for the target series are based on a set of input variables. The forecast module is trained by varying θ to minimize forecast error (typically mean squared error), which is an *intermediate quantity*. A more direct approach would be to simultaneously vary θ and θ' to maximize a measure of ultimate performance $U(\theta, \theta')$, such as trading profits, utility or risk-adjusted return. Note that the trading module typically does not make use of the inputs used by the forecast module, resulting in a loss of information or a *forecast bottleneck*. Performance of such a system is thus likely to be suboptimal

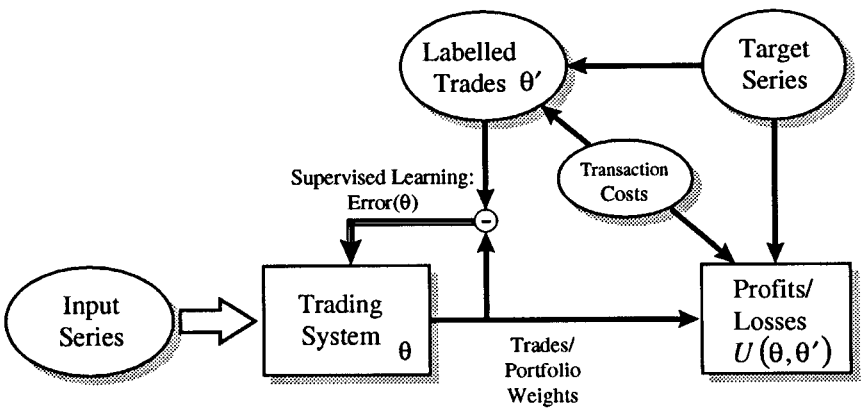

Figure 2. A trading system trained with labelled data. The system includes a trading module with parameters θ and a labelling procedure with parameters θ'. Trades are based on a set of input variables. Target trades are produced by the labelling procedure. The trading module is trained on the labelled trades using a supervised learning approach to vary θ. The ultimate performance of the system depends upon how good the labelling algorithm is (as determined by θ'), and how well the trading module can learn to trade (by varying θ) using the input variables and labelled trades. Since the ultimate measure of performance $U(\theta, \theta')$ is not used to optimize θ directly, performance of such a system is thus likely to be suboptimal

labelled data attempts to solve the *structural credit assignment* problem.* Certain difficulties arise when trying to solve the structural and temporal credit assignment problems separately in this way, particularly when transaction costs are included.

The performance achievable in practice by the trading module will usually be substantially worse than that suggested by the labelled trades. This is because most labelling procedures are based on only the target series (possibly taking into account transaction costs), ignore the input variables and do not consider the conditional distributions of price changes in the target series given the input variables. Moreover, since transactions costs depend upon the actual sequence of trades made, the simulated costs associated with the labelled trades will differ from those incurred in practice. Hence, a labelling procedure is not likely to give rise to a sequence of trades that is realizable in practice or to a realistic assessment of the actual transaction costs likely to occur. Finally, since $U(\theta, \theta')$ is not optimized directly (see Figure 2), supervised learning based on labelled data will yield suboptimal performance.

Direct optimization of performance via recurrent reinforcement learning

A trading system can be optimized to solve both the temporal credit assignment and structural credit assignment problems mentioned above *simultaneously* using reinforcement learning (see, for example, Sutton and Barto, 1997 and references therein). We adopt this approach here.

Reinforcement learning algorithms find approximate solutions to stochastic dynamic programming problems (Bertsekas, 1995) and can do so in an on-line mode. In reinforcement learning, target outputs are not provided. Rather, the system takes actions (makes trades), receives feedback on its performance (an evaluation signal) and then adjusts its internal parameters to increase its future rewards. With this approach, an ultimate measure of trading performance $U(\theta)$, such as profit, utility or risk-adjusted return is optimized directly. See Figure 3.

A simultaneous solution of the structural and temporal credit assignment problems will generally require using a *recurrent* learning algorithm. Trading system profits depend upon sequences of interdependent decisions, and are thus path-dependent. Optimal trading decisions when the effects of transactions costs, market impact and taxes† are included require knowledge of the current system state. Including information related to past decisions in the inputs to a trading system results in a *recurrent* decision system.‡ The proper optimization of a recurrent, path-dependent decision system is quite different from the simple supervised optimization techniques used for direct forecasts or for labelled trading data.

Reinforcement learning analogs of recurrent learning algorithms are required to train our proposed systems. Such recurrent learning algorithms include both off-line (batch) training algorithms like backpropagation through time (BPTT) (Rumelhart, Hinton and Williams, 1986; Werbos, 1990) or on-line (adaptive) algorithms such as real-time recurrent learning (RTRL) (Williams and Zipser, 1989) or dynamic backpropagation (DBP) (Narendra and Parthasarathy, 1990). The recurrent reinforcement learning algorithms that we utilize here are novel, but

* This terminology was proposed in Sutton (1988).

† For brevity, we omit further discussion of market impact and tax effects.

‡ Here, recurrence refers to the nature of the algorithms required to optimize the system. For example, optimizing a feedforward NAR(p) model for one-step-ahead prediction does not require a recurrent learning algorithm, while optimizing the same NAR(p) model to perform iterated predictions *does*. The fact that a forecast or decision is made by a feedforward, non-recurrent network does not mean that optimizing it correctly can be done with a standard, non-recurrent training procedure.

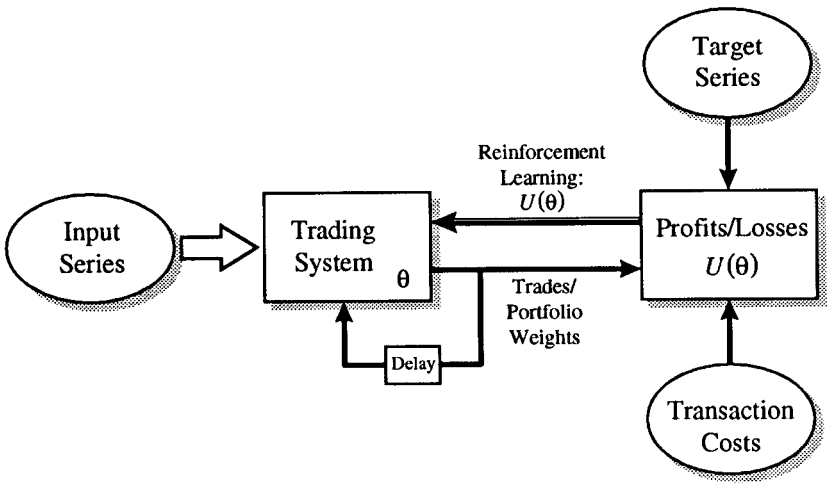

Figure 3. A trading system based on recurrent reinforcement learning, the approach taken in this paper. The system makes trading decisions directly based upon a set of input variables and the system state. A trading performance function $U(\theta)$, such as profit, utility or risk-adjusted return, is used to *directly optimize* the trading system parameters θ using reinforcement learning. The system is recurrent; the feedback of system state (current positions or portfolio weights) enables the trading system to learn to correctly incorporate transactions costs into its trading decisions. In comparison to the systems in Figures 1 and 2, no intermediate steps such as making forecasts or labelling desired trades are required

straightforward. They are variants of the above mentioned algorithms that maximize immediate rewards in an on-line fashion. We refer to them as RRL algorithms.

Related work

Several papers that relate to ours have recently appeared. Samuelson (1990) stimulated our interest in power law utility functions and their relation to the differential Sharpe ratio. White (1996) suggested using a performance ratio based on the second lower partial moment to us. We became aware of the other references listed below after developing our basic approach.

Samuelson (1990) uses logarithmic and power law utility functions, which we use in the third section to evaluate simple asset allocation or market timing strategies (which he calls 'across-time diversification'). Samuelson's analysis shows that under the assumption of a random walk price process, the optimum behaviour for a trader with a power law utility function is to hold constant proportions of risky and risk-free securities. That is, in the absence of superior forecasting ability, across-asset-class diversification will on a risk-adjusted basis outperform across-time diversification. In contrast, our approach assumes that superior forecasting and trading strategies are not impossible, and that dynamic asset allocation strategies may in some cases achieve higher utility than simple fixed allocation methods.

Timmermann and Pesaran (1995) use wealth and the Sharpe ratio as selection criteria (rather than optimization criteria) for trading systems. The set of traders considered are based on linear forecasting models that differ in the subsets of input variables included. The forecasting models are linear regressions with parameters estimated to minimize mean squared forecast

error (MSFE). The wealth and the Sharpe ratio performance functions are not used for direct optimization of system parameters. The selection among forecasting models is updated periodically. The authors are able to use their simulation results to document predictability in monthly US stock returns. In related work, Satchell and Timmermann (1995) provide arguments that MSFE is a bad indicator for potential trading profits. They prove a theorem that there is not necessarily any monotonic relationship between the size of the MSFE and the probability of correctly forecasting the sign of a variable.

In independent work, Bengio (1997) points out that global optimization of trading systems consisting of separate forecasting and trading modules (such as that shown in Figure 1) provides better results than separately minimizing MSFE of the forecast module and subseqently maximizing profit of the trading module. Bengio optimizes portfolios and employs back-propagation through time to maximize final wealth. Kang, Choey and Weigend (1997) compare optimization of the Sharpe ratio and profits using a non-recursive supervised learning method applied to a simple asset allocation strategy. The allocation that is varied is the amount of capital invested in a fixed trading strategy. However, the authors do not directly optimize the trading system parameters (those that determine when to buy or sell) or take into account transaction costs. Neuneier (1996) uses Q-Learning (Watkins, 1989; Watkins and Dayan, 1992) to train an asset-allocation system to maximize profit. Transaction costs consisting of both fixed and proportional parts are included in the analysis. Simulation results are presented for an artificial exchange rate trading system and a system that switches between cash and a portfolio of German stocks that tracks the DAX. The description of the methods used is sketchy, and most relevant details of the empirical work are not disclosed. Finally, White (1996) has done simulations that optimize the Sharpe ratio and other measures or risk-adjusted return based on downside risk (see below).

STRUCTURE AND OPTIMIZATION OF TRADERS AND PORTFOLIOS

Structure and optimization of traders

Single asset with discrete position size

In this section, we consider performance functions for systems that trade a single risky security with price series z_t. We also consider a risk-free bond whose rate of return is known one time step ahead, which has cumulative price series z_t^f. Since the trader can realize returns from either the risky security or the bond, the trader is actually making asset allocation decisions.

The trader is assumed to take only short, neutral or long positions $F_t \in \{-1, 0, 1\}$ of constant magnitude in the risky security z_t. A conservative strategy for stock or bond investments might be restricted to $F_t \in \{0, 1\}$, while a simplified reversal trading strategy could have no neutral state, $F_t \in \{-1, 1\}$. The constant magnitude assumption can be easily relaxed to enable better risk control. The position F_t is established or maintained at the end of each time interval t, and is reassessed at the end of period $t + 1$. A trade is thus possible at the end of each time period, although non-zero trading costs will discourage excessive trading. A trading system return R_t is realized at the end of the time interval $(t - 1, t]$ and includes the profit or loss resulting from the position F_{t-1} held in the risky security during that interval, any returns from positions in the risk-free bond, and any transaction cost incurred at time t is due to a difference in the positions F_{t-1} and F_t.

In order to properly incorporate the effects of transactions costs, market impact and taxes in a trader's decision making, the trader must have internal state information and must therefore be

recurrent. An example of a single asset trading system that could take into account transactions costs and market impact would be one with the following decision function:*

$$F_t = F(\theta_t; F_{t-1}, I_t) \text{ with } I_t = \{z_t, z_{t-1}, z_{t-2}, \ldots; z^f_{t+1}, z^f_t, z^f_{t-1}, \ldots; y_t, y_{t-1}, y_{t-2}, \ldots\} \quad (1)$$

where θ_t denotes the (learned) system parameters at time t and I_t denotes the information set at time t, which includes present and past values of the price series z_t and z^f_t and an arbitrary number of other external variables denoted y_t. More general decision functions could include the current trade's profit/loss (for capturing capital gains tax effects), past positions F_{t-s} and past profits and losses (in analogy with the moving average components of ARIMA models) and other factors.

Optimizing profit and wealth

Trading systems can be optimized by maximizing performance functions $U()$ such as profit, wealth, utility functions of wealth or performance ratios like the Sharpe ratio. The Sharpe and other performance ratios will be discussed below. The simplest and most natural performance function for a risk-insensitive trader is profit. We consider two cases: additive and multiplicative profits. The transactions cost rate (per share, per contract, or per dollar amount, depending on context) is denoted δ.

Additive profits are appropriate to consider if each trade is for a fixed number of shares or contracts of security z_t. This is often the case, for example, when trading small futures accounts or when trading standard US$ FX contracts in dollar-denominated foreign currencies. With the definitions $r_t = z_t - z_{t-1}$ and $r^f_t = z^f_t - z^f_{t-1}$ for the price returns of a risky (traded) asset and a risk-free asset (like T-bills) respectively, the additive profit accumulated over T time periods with trading position size $\mu > 0$ is then defined as:

$$P_T = \mu \sum_{t=1}^{T} R_t = \mu \sum_{t=1}^{T} \{r^f_t + F_{t-1}(r_t - r^f_t) - \delta \,|\, F_t - F_{t-1} \,|\,\} \quad (2)$$

where $P_0 = 1$ and typically $F_T = F_0 = 0$. We have implicitly defined the return for the trade completed at time t as R_t. Note that the transaction costs for switching between short and long positions are twice those for switching between neutral and long/short positions. Equation (2) holds for continuous quantities also. The wealth is defined as $W_T = W_0 + P_T$.

Multiplicative profits are appropriate when a fixed fraction of accumulated wealth $\nu > 0$ is invested in each long or short trade. Here, $r_t = (z_t/z_{t-1} - 1)$ and $r^f_t = (z^f_t/z^f_{t-1} - 1)$. If no short sales are allowed and the leverage factor is set fixed at $\nu = 1$, the wealth at time T is:

$$\begin{aligned} W_T &= W_0 \prod_{t=1}^{T} \{1 + R_t\} \\ &= W_0 \prod_{t=1}^{T} \{1 + (1 - F_{t-1}) r^f_t + F_{t-1} r_t\} \{1 - \delta \,|\, F_t - F_{t-1} \,|\,\} \end{aligned} \quad (3)$$

* Note that the value of the risk free bond one step ahead z^f_{t+1} is known at time t.

where R_t is the return realized for the period ending at time t. In this case, the profit is $P_T = W_T - W_0$. If short sales or leverage $\nu \neq 1$ are allowed, then the correct expression depends in detail on the timing of the sequence of trades. For brevity, we omit the discussion for this case.

Structure and optimization of portfolios

Portfolios: continuous quantities of multiple assets

When the risk-free rate of return r_t^f is included in single risky-asset trading models as above, one actually has a simple two-asset portfolio. For trading multiple assets in general (typically including a risk-free instrument), a multiple output trading system is required. Denoting a set of m markets with price series $\{\{z_t^a\} : a = 1, \ldots, m\}$, the market return r_t^a for price series z_t^a for the period ending at time t is defined as $(z_t^a/z_{t-1}^a - 1)$. Defining portfolio weights of the ath asset as $F^a(\ )$, a trader that takes only long positions must have portfolio weights that satisfy:

$$F^a \geqslant 0 \quad \text{and} \quad \sum_{a=1}^{m} F^a = 1 \tag{4}$$

With these constraints, standard Markowitz mean-variance portfolio optimization is a quadratic programming problem. However, when optimizing the parameters of a non-linear trading system, portfolio optimization becomes a non-linear programming problem.

One approach to imposing the constraints on the portfolio weights (4) without requiring that a constained optimization be performed is to use a trading system that has softmax outputs:

$$F^a(\,) = \frac{\exp[f^a(\,)]}{\sum_{b=1}^{m} \exp[f^b(\,)]} \quad \text{for } a = 1, \ldots, m \tag{5}$$

Here, the $f^a(\,)$ could be linear or more complex functions of the inputs, such as a two-layer neural network with sigmoidal internal units and linear outputs. Such a trading system can be optimized using unconstrained optimization methods. Note, however, that the portfolio weights F^a obtained are invariant under shifts in the values of the f^a of the form $\{f^a \to f^a + c; a = 1, \ldots, m\}$, so multiple solutions for the f^a exist. Denoting the sets of raw and normalized outputs collectively as the vectors $\mathbf{f}(\,)$ and $\mathbf{F}(\,)$ respectively, a recursive trader will have structure

$$\mathbf{F}_t = \text{softmax}\{\mathbf{f}_t(\theta_{t-1}; \mathbf{F}_{t-1}, I_t)\} \tag{6}$$

Profit and wealth for portfolios

When multiple assets are considered, the effective portfolio weightings change with each time step due to price movements. Thus, maintaining constant or desired portfolio weights requires that adjustments in positions be made at each time step. The wealth after T periods for a portfolio trading system is

$$\begin{aligned} W_T &= W_0 \prod_{t=1}^{T} \{1 + R_t\} \\ &= W_0 \prod_{t=1}^{T} \left\{ \left(\sum_{a=1}^{m} F_{t-1}^a \frac{z_t^a}{z_{t-1}^a} \right) \left(1 - \delta \sum_{a=1}^{m} | F_t^a - \tilde{F}_t^a | \right) \right\} \end{aligned} \tag{7}$$

where $\tilde{F}_t^a$ is the effective portfolio weight of asset a before readjusting, defined as

$$\tilde{F}_t^a = \frac{F_{t-1}^a (z_t^a / z_{t-1}^a)}{\sum_{b=1}^{m} F_{t-1}^b (z_t^b / z_{t-1}^b)} \tag{8}$$

and we have defined the trading returns R_t implicitly. In equation (7), the first factor in the curly brackets is the increase in wealth over the time interval t prior to rebalancing to achieve the newly specified weights F_t^a. The second factor is the reduction in wealth due to the rebalancing costs. The profit after T periods is $P_T = W_T - W_0$.

Optimization of traders and portfolios via recurrent reinforcement learning

Reinforcement learning algorithms yield approximate solutions to stochastic dynamic programming problems and are able to do so on-line (see, for example, Bertsekas, 1995).

Reinforcement learning adjusts the parameters of a system to maximize the expected payoff or reward that is generated due to the actions of the system. This is accomplished through trial and error exploration of the environment. Unlike supervised learning, the system is not presented with examples of desired strategies. Rather, it receives a reinforcement signal from its environment (a *reward*) that provides information on whether its actions are good or bad.

The structural credit assignment problem refers to the problem of assigning credit to the individual parameters of a system. If the reward produced also depends on a series of actions of the system, then the temporal credit assignment problem is encountered, i.e. assigning credit to the individual actions of the system through time. Reinforcement learning algorithms solve both problems simultaneously.

The performance functions that we consider are functions of profit or wealth $U(W_T)$ after a sequence of T time steps, or more generally of the whole time sequence of trades $U(W_1, W_2, \ldots, W_t, \ldots, W_T)$. The simple form $U(W_T)$ includes the economic utility functions discussed in the next section. The second case is the general form for path-dependent performance functions like the Sharpe ratio and Sterling ratio. In either case, the performance function at time T can be expressed as a function of the sequence of trading returns $U(R_1, R_2, \ldots, R_t, \ldots, R_T)$. We denote this by U_T in the rest of this section.

Given a trading system model $F_t(\theta)$, the goal is to adjust the parameters θ in order to maximize U_T. This maximization for a complete sequence of T trades can be done off-line using dynamic programming or batch versions of reinforcement learning algorithms. Alternatively, the optimization can be done on-line using standard reinforcement learning techniques. Most of our simulations make use of on-line approaches based on stochastic gradient ascent, since they offer advantages in computational efficiency.*

The gradient of U_T with respect to the parameters θ of the system after a sequence of T trades is

$$\frac{\mathrm{d}U_T(\theta)}{\mathrm{d}\theta} = \sum_{t=1}^{T} \frac{\mathrm{d}U_T}{\mathrm{d}R_t} \left\{ \frac{\mathrm{d}R_t}{\mathrm{d}F_t} \frac{\mathrm{d}F_t}{\mathrm{d}\theta} + \frac{\mathrm{d}R_t}{\mathrm{d}F_{t-1}} \frac{\mathrm{d}F_{t-1}}{\mathrm{d}\theta} \right\} \tag{9}$$

* The use of stochastic optimization methods is widely viewed in the adaptive signal processing, neural network and machine learning communities as more computationally efficient than other batch mode non-linear optimization methods.

The above expression as written with scalar F_t can be applied to traders of a single risky asset described above by backpropagating the reinforcement signal through the pre-thresholded outputs in a manner similar to the Adaline learning rule (Widrow and Hoff, 1960). The expression can also be trivially generalized to the vector case for portfolios, as described above.

The system can be optimized in batch mode by repeatedly computing the value of U_T on forward passes through the data and adjusting the trading system parameters by using grading ascent (with learning rate ρ)

$$\Delta\theta = \rho \frac{\mathrm{d}U_T(\theta)}{\mathrm{d}\theta} \tag{10}$$

or some other optimization method. Note that the quantities $\mathrm{d}F_t/\mathrm{d}\theta$ are *total derivatives* that depend upon the entire sequence of previous trades. To correctly compute and optimize these total derivatives requires that a reinforcement version of a recurrent algorithm like BPTT (Rumelhart *et al.*, 1986; Werbos, 1990), RTRL (Williams and Zipser, 1989) or dynamic backpropagation (Narendra and Parthasarathy, 1990) be used. These algorithms account for the temporal dependencies in a sequence of decisions through a recursive update equation for the parameter gradients:

$$\frac{\mathrm{d}F_t}{\mathrm{d}\theta} = \frac{\partial F_t}{\partial \theta} + \frac{\partial F_t}{\partial F_{t-1}} \frac{\mathrm{d}F_{t-1}}{\mathrm{d}\theta} \tag{11}$$

A simple on-line stochastic optimization can be obtained by considering only the term in equation (9) that depends on the most recently realized return R_t during a forward pass through the data:

$$\frac{\mathrm{d}U_t(\theta)}{\mathrm{d}\theta} = \frac{\mathrm{d}U_t}{\mathrm{d}R_t} \left\{ \frac{\mathrm{d}R_t}{\mathrm{d}F_t} \frac{\mathrm{d}F_t}{\mathrm{d}\theta} + \frac{\mathrm{d}R_t}{\mathrm{d}F_{t-1}} \frac{\mathrm{d}F_{t-1}}{\mathrm{d}\theta} \right\} \tag{12}$$

Note that this equation is correct if U_T is the sum of the individual U_T's; otherwise, it is an approximation. The parameters are then updated on-line using

$$\Delta\theta_t = \rho \frac{\mathrm{d}U_t(\theta_t)}{\mathrm{d}\theta_t} \tag{13}$$

Such an algorithm performs a stochastic optimization (since the system parameters θ_t are varied *during* each forward pass through the training data), and is an example of *immediate reward* reinforcement learning as it only performs one-step ahead optimization. The stochastic, on-line analog of equation (11) is:

$$\frac{\mathrm{d}F_t}{\mathrm{d}\theta_t} \approx \frac{\partial F_t}{\partial \theta_t} + \frac{\partial F_t}{\partial F_{t-1}} \frac{\mathrm{d}F_{t-1}}{\mathrm{d}\theta_{t-1}} \tag{14}$$

We use on-line algorithms of this recurrent reinforcement learning (RRL) type in the simulations presented in the fifth section. The fourth section develops an on-line approach for optimizing a *differential* Sharpe ratio. To our knowledge, the RRL approach that we present here is unique.

Alternative reinforcement learning approaches include *delayed reward* algorithms, examples of which include the actor-critic method (Barto, Sutton and Anderson, 1983) and Q-Learning (Watkins, 1989; Watkins and Dayan, 1992). We compare Q-Learning to our recurrent reinforcement learning algorithms in Moody *et al*. (1998).

OPTIMIZING ECONOMIC UTILITY

The optimization of profit and wealth described above assumes that investors are insensitive to risk. However, most investors are more sensitive to losses than to gains, and are therefore willing to sacrifice some potential gains in order to have less risk of loss. There is also an intrinsic asymmetry between percentage losses and gains: a 25% loss must be followed by 33% gain in order to break even. Utility functions of wealth $U(W)$ can capture various kinds of risk/reward preferences.

In 1738, Daniel Bernoulli proposed the logarithmic utility function $U(W_t) = \log W_t$ (see Bernoulli, 1954). A more general class of utility functions that capture varying degrees of risk sensitivity are the power law utilities:

$$\begin{aligned} U_\gamma(W_t) &= W_t^\gamma/\gamma \quad \text{for } \gamma \neq 0 \\ U_0(W_t) &= \log W_t \quad \text{for } \gamma = 0 \end{aligned} \tag{15}$$

These utility functions have constant relative risk aversion, defined as

$$\mathcal{R}(W) = -\frac{\mathrm{d} \log U'(W)}{\mathrm{d} \log W} = 1 - \gamma \tag{16}$$

The case $\gamma = 1$ is risk-neutral, and $U_1(W)$ corresponds to absolute wealth or profit (equation (2)) while utilities with $\gamma > 1$ describe risk-seeking or 'thrill-seeking' behaviour. Most investors have risk-averse utility functions ($\gamma < 1$), with smaller values of γ corresponding to greater sensitivity to loss. The limit $\gamma \to -\infty$ corresponds to absolute risk aversion.

The sensitivity to risk on a per time period or per trade basis can be seen by considering a second-order Taylor expansion of the contribution of R_t to $U_\gamma(W_T)$:

$$\begin{aligned} U_\gamma(1 + R_t) &\approx \frac{1}{\gamma} + R_t + \frac{\gamma - 1}{2} R_t^2 \text{ for } \gamma \neq 0 \\ U_\gamma(1 + R_t) &\approx 0 + R_t - \frac{1}{2} R_t^2 \text{ for } \gamma = 0 \end{aligned} \tag{17}$$

Taking the expectation value of $U_\gamma(1 + R_t)$, and defining risk to be $E(R_t^2)$, we see that for $\gamma > 1$, risk is positively weighted, while for $\gamma < 1$, risk is negatively weighted. Note that the quantity $\lambda \equiv \mathcal{R}(W) = 1 - \gamma$ is called the *coefficient of risk aversion*, so the risk {averse, neutral, seeking} cases have $\{\lambda > 0, \lambda = 0, \lambda < 0\}$ respectively.

THE SHARP RATIO AND THE *DIFFERENTIAL* SHARPE RATIO

Optimizing the Sharpe ratio

The Sharpe ratio is a measure of risk-adjusted return (Sharpe, 1966, 1994). Denoting as before the trading system returns for period t (including transactions costs) as R_t, the Sharpe ratio is defined to be

$$S_T = \frac{\text{Average}(R_t)}{\text{Standard deviation}(R_t)} \tag{18}$$

where the average and standard deviation are estimated over returns for periods $t = \{1, \ldots, T\}$. A trading system can be trained to maximize the Sharpe ratio as follows. First, we define the Sharpe ratio for n returns R_i in terms of estimates of the first and second moments of the returns' distributions:

$$S_n = \frac{A_n}{K_n(B_n - A_n^2)^{1/2}} \tag{19}$$

with

$$A_n = \frac{1}{n}\sum_{i=1}^{n} R_i \quad B_n = \frac{1}{n}\sum_{i=1}^{n} R_i^2 \quad K_n = \left(\frac{n}{n-1}\right)^{1/2} \tag{20}$$

The normalizing factor K_n is required for an unbiased estimate of the standard deviation, but is not relevant for optimization. The derivative with respect to the system parameters (using scalar notation for θ, even though it is generally a vector) is:

$$\begin{aligned} \frac{\mathrm{d}S_n(\theta)}{\mathrm{d}\theta} &= \sum_{i=1}^{n}\left\{\frac{\mathrm{d}S}{\mathrm{d}A_n}\frac{\mathrm{d}A_n}{\mathrm{d}R_i} + \frac{\mathrm{d}S}{\mathrm{d}B_n}\frac{\mathrm{d}B_n}{\mathrm{d}R_i}\right\}\left\{\frac{\mathrm{d}R_i}{\mathrm{d}F_i}\frac{\mathrm{d}F_i}{\mathrm{d}\theta} + \frac{\mathrm{d}R_i}{\mathrm{d}F_{i-1}}\frac{\mathrm{d}F_{i-1}}{\mathrm{d}\theta}\right\} \\ &= \frac{1}{n}\sum_{i=1}^{n}\left\{\frac{B_n - A_n R_i}{K_n(B_n - A_n^2)^{3/2}}\right\}\left\{\frac{\mathrm{d}R_i}{\mathrm{d}F_i}\frac{\mathrm{d}F_i}{\mathrm{d}\theta} + \frac{\mathrm{d}R_i}{\mathrm{d}F_{i-1}}\frac{\mathrm{d}F_{i-1}}{\mathrm{d}\theta}\right\} \end{aligned} \tag{21}$$

The above expression is written with scalar F_i applies to the traders of a single risky asset described above but can be trivially generalized to the vector case for portfolios, as described earlier.

The system can be optimized in batch mode by repeatedly computing the value of S_n on forward passes through the data and adjusting the trading system parameters by using gradient ascent (with learning rate ρ)

$$\Delta\theta_n = \rho\frac{\mathrm{d}S_n(\theta)}{\mathrm{d}\theta} \tag{22}$$

or some other optimization method. A simple incremental optimization might consider only the term in equation (21) dependent on the last realized return R_n. Note that the quantities $\mathrm{d}F_i/\mathrm{d}\theta$ are *total derivatives* that depend upon the entire sequence of previous trades. To correctly

compute and optimize these total derivatives requires that a recurrent algorithm like BPTT, RTRL or dynamic backpropagation be used.

Running and exponential moving average Sharpe ratios

In order to facilitate on-line learning, an incremental Sharpe ratio is required. First, we define a *running Sharpe ratio* by making use of recursive estimates of the first and second moments of the returns' distributions:

$$A_n = \frac{1}{n} R_n + \frac{n-1}{n} A_{n-1} \quad \text{and} \quad B_n = \frac{1}{n} R_n^2 + \frac{n-1}{n} B_{n-1} \tag{23}$$

(with $A_0 = B_0 = 0$) in the evaluation of equation (19). Next, we extend this definition to an *exponential moving average Sharpe ratio* on time scale η^{-1} by making use of moving average estimates of the first and second moments of the returns' distributions:

$$S_t = \frac{A_t}{K_\eta (B_t - A_t^2)^{1/2}} \tag{24}$$

with

$$\begin{aligned} A_t &= \eta R_t + (1-\eta) A_{t-1} \\ B_t &= \eta R_t^2 + (1-\eta) B_{t-1} \\ K_\eta &= \left(\frac{1-\eta/2}{1-\eta} \right)^{1/2} \end{aligned} \tag{25}$$

initialized with $A_\eta(0) = B_\eta(0) = 0$. In equation (24), the normalization factor K_η is required for an unbiased estimate of the moving standard deviation. For purposes of trading system optimization, however, this constant factor can be ignored.

***Differential* Sharpe ratios for on-line optimization**

While both the running and moving Sharpe ratios can be used to optimize trading systems in batch or off-line mode, proper on-line learning requires that we compute the influence on the Sharpe ratio of the return at time t. With the running or moving Sharpe ratios defined above, we can derive *differential Sharpe ratios* for on-line optimization of trading system performance. It is advantageous to use on-line performance functions both to speed the convergence of the learning process (since parameter updates can be done *during* each forward pass through the training data) and to adapt to changing market conditions during live trading. The update equations for the exponential moving estimates can be rewritten

$$\begin{aligned} A_t &= A_{t-1} + \eta \Delta A_t = A_{t-1} + \eta (R_t - A_{t-1}) \\ B_t &= B_{t-1} + \eta \Delta B_t = B_{t-1} + \eta (R_t^2 - B_{t-1}) \end{aligned} \tag{26}$$

where we have implicitly defined the update quantities ΔA_t and ΔB_t. Treating A_{t-1}, B_{t-1} and K_η as numerical constants, note that η in the update equations (26) controls the magnitude of the

influence of the return R_t on the Sharpe ratio S_t. Taking $\eta \to 0$ turns off the updating, and making η positive turns it on.

With this in mind, we can obtain a *differential* Sharpe ratio by expanding equation (24) to first order in η:*

$$S_t \approx S_{t-1} + \eta \frac{\mathrm{d}S_t}{\mathrm{d}\eta} \mid_{\eta=0} + O(\eta^2) \tag{27}$$

Noting that only the first-order term in this expansion depends upon the return R_t at time t (through ΔA_t and ΔB_t), we define the *differential Sharpe ratio* as:

$$D_t \equiv \frac{\mathrm{d}S_t}{\mathrm{d}\eta} = \frac{B_{t-1}\Delta A_t - \frac{1}{2}A_{t-1}\Delta B_t}{(B_{t-1} - A_{t-1}^2)^{3/2}} \tag{28}$$

The current return R_t enters this expression only in the numerator through $\Delta A_t = R_t - A_{t-1}$ and $\Delta B_t = R_t^2 - B_{t-1}$.

The influences of risk and return on the differential Sharpe ratio are readily apparent. The first term in the numerator is positive if R_t exceeds the moving average of past returns A_{t-1}, while the second term is negative if R_t^2 exceeds the moving average of past squared returns B_{t-1}. Assuming that $A_{t-1} > 0$, the largest possible improvement in D_t occurs when

$$R_t^* = B_{t-1}/A_{t-1} \tag{29}$$

Thus, the Sharpe ratio actually penalizes returns larger than R_t^*, which is counter-intuitive relative to most investors' notions of risk and reward.

Note that a second expression for a differential Sharpe ratio can be obtained by considering a perturbative expansion of the current return R_t relative to the exponential moving average of past returns A_{t-1}. This is obtained by expanding S_t of equation (24) in a Taylor series about $R_t = A_{t-1}$ to second order in ΔA_t, and expressing the coefficients in terms of information available at time $t-1$. The result is:

$$\begin{aligned} S_t &\approx \frac{1}{(1-\eta)^{1/2}} \left\{ S_{t-1} + \frac{\eta}{K_\eta} D_t + O((\Delta A_t)^3) \right\} \\ D_t &\equiv \frac{(B_{t-1} - A_{t-1}^2)\Delta A_t - \frac{1}{2}A_{t-1}(\Delta A_t)^2}{(B_{t-1} - A_{t-1}^2)^{3/2}} \end{aligned} \tag{30}$$

The numerators of equations (28) and (30) differ only by a constant. As for equation (28), this expression too is maximized when $R_t^* = B_{t-1}/A_{t-1}$.

The differential Sharpe ratio D_t is used in equation (12) as the current contribution to the performance function U_t. Since S_{t-1} in equation (27) does not depend on R_t, we have

* Note that $S_t \to S_{t-1}$ as $\eta \to 0$. Again, we treat A_{t-1}, B_{t-1} and K_η as constants. As a technical point, differentiation of K_η with respect to η can be avoided by considering an expansion with terms $K_\eta^{-1}\mathrm{d}^m(K_\eta S_t)/\mathrm{d}\eta^m$.

$\mathrm{d}U_t/\mathrm{d}R_t = \mathrm{d}S_t/\mathrm{d}R_t \approx \eta \mathrm{d}D_t/\mathrm{d}R_t$. Whether optimizing a trading system with equations (28) or (30), the relevant derivatives have the same simple form:

$$\frac{\mathrm{d}D_t}{\mathrm{d}R_t} = \frac{B_{t-1} - A_{t-1}R_t}{(B_{t-1} - A_{t-1}^2)^{3/2}} \tag{31}$$

Advantages of the differential Sharpe ratio

The differential Sharpe ratio has several attractive properties:

- *Facilitates recursive updating*: The incremental nature of the calculations of A_t and B_t make updating the exponential moving Sharpe ratio straightforward. It is not necessary to recompute the average and standard deviation of returns for the entire trading history in order to update the Sharpe ratio for the most recent time period.
- *Enables efficient on-line optimization*: D_t and $\mathrm{d}D_t/\mathrm{d}R_t$ can be cheaply calculated using the previously computed moving averages A_{t-1} and B_{t-1} and the current return R_t. This enables efficient stochastic optimization without the need to compute complete sums as in equation (21).
- *Weights recent returns more*: Based on the exponential moving average Sharpe ratio, recent returns receive stronger weightings in D_t than do older returns.
- *Provides interpretability*: The differential Sharpe ratio isolates the contribution of the current return R_t to the exponential moving average Sharpe ratio. The simple form of D_t makes clear how risk and reward affect the Sharpe ratio. The relationship of the Sharpe ratio to economic utility functions is also readily apparent. (See next section.)

Relation of Sharpe ratio to economic utility functions

The Sharpe ratio is not a standard economic utility function $U(W_T)$, since its value depends not just on current wealth (or current changes in wealth) but also on past performance. It is an example of a path-dependent performance function $U(W_1, W_2, \ldots, W_t, \ldots, W_T)$ or $U(R_1, R_2, \ldots, R_t, \ldots, R_T)$. This can be seen by comparing the numerator of the differential Sharpe ratio (28)

$$B_{t-1}\left\{-\frac{1}{2}A_{t-1} + R_t - \frac{A_{t-1}}{2B_{t-1}}R_t^2\right\} \tag{32}$$

with equation (17). In a sense, the Sharpe ratio can be thought of as an adaptive utility function, since the relative weightings of risk and return depend on the performance history as measured by A_t and B_t. The effective coefficient of risk aversion at time t is $\lambda_t = A_{t-1}/B_{t-1}$. Note that $\lambda_t = R_t^{*-1}$.

Other performance ratios

Another measure related to the Sharpe ratio is the *appraisal ratio* (Treynor and Black, 1973) or the *information ratio* (Grinold and Kahn, 1995), which measures risk and return relative to some benchmark. It is defined as

$$\text{Information ratio} = \frac{\text{(Annualized) residual return}}{\text{(Annualized) residual risk}} \tag{33}$$

and is commonly used to measure the performance of active portfolio management strategies relative to an index. Our methods can be easily generalized to maximize such quantities.

Since the standard Sharpe ratio normalizes average excess returns by the standard deviation of excess returns, it does not properly distinguish between upside potential and downside risk. As discussed above, the differential Sharpe ratio actually penalizes returns larger than R_t^*. Two alternative performance functions normalize average returns by the *semi-variance* (Markowitz, 1959) or the *downside deviation* or *second lower partial moment* (SLPM) (Nawrocki, 1991, 1992; Sortino and Vandermeer, 1991; Sortino and Forsey, 1996; White, 1996). Estimates of these performance ratios can be defined as

$$I_{1n} = \frac{A_n}{\hat{\sigma}_n^-} \text{ with } (\hat{\sigma}_n^-)^2 = \frac{1}{n-1}\sum_{i=1}^{n}(\min\{(R_i - A_n), 0\})^2 \tag{34}$$

$$I_{2n} = \frac{A_n}{\hat{s}_n^-} \text{ with } (\hat{s}_n^-)^2 = \frac{1}{n}\sum_{i=1}^{n}(\min\{R_i, 0\})^2 \tag{35}$$

The semi-variance gives non-zero weight to below-average returns, even if positive, while the downside deviation gives non-zero weight only to negative excess returns. I_{2n}, based on the downside deviation, tends to be well correlated (White, 1996) with the *Sterling ratio*, a trading system performance function widely used in the fund management community. The Sterling ratio is commonly defined as:

$$\text{Sterling ratio} = \frac{\text{Average return per period (annualized)}}{\text{Maximum peak to trough drawn-down}} \tag{36}$$

Extensions of both equations (34) and (35) to running, moving and differential performance ratios as developed above for the Sharpe ratio are straightforward.

EMPIRICAL RESULTS

Trader simulation

Data

We generate log price series as random walks with autoregressive trend processes. The two-parameter model is thus:

$$p(t) = p(t-1) + \beta(t-1) + k\varepsilon(t) \tag{37}$$

$$\beta(t) = \alpha\beta(t-1) + \nu(t) \tag{38}$$

where α and k are constants, and $\varepsilon(t)$ and $\nu(t)$ are normal random deviates with zero mean and unit variance. We define the artificial price series as

$$z(t) = \exp\left(\frac{p(t)}{R}\right) \tag{39}$$

where R is a scale defined as the range of $p(t)$: $\max(p(t)) - \min(p(t))$ over a simulation with 10,000 samples.*

* This is slightly more than the number of hours in a year (8760), so the series could be thought of as representing hourly prices in a 24-hour artificial market. Alternatively, a series of this length could represent slightly less than five years of hourly data in a market that trades about 40 hours per week.

Long/short trader of single security

We have tested techniques for optimizing both profit and the Sharpe ratio in a variety of settings. In this section, we present results for optimizing three simple trading systems using data generated for an artificial market. The systems are long/short trading systems with recurrent state similar to that described above. Two of the systems are trained via recurrent reinforcement learning to maximize the differential Sharpe ratio (28) or profit. These two systems are compared to a third trading system built on a forecasting system that minimizes forecast MSE.

For the results we present here, we set the parameters of the price series to $\alpha = 0.9$ and $k = 3$. The artificial price series are trending on short time scales and have a high level of noise. An example of the artificial price series is shown in the top panel of Figure 4.

In our simulations we find that maximizing the differential Sharpe ratio yields more consistent results than maximizing profits, and that both methods outperform the trading system based on forecasts.

Simulated trading results

Figures 4, 5 and 6 show results for a single simulation for an artificial market as described above. The trading system is initialized randomly at the beginning, and adapted using real-time recurrent learning to optimize the differential Sharpe ratio (28). The transaction costs are fixed at a half percent during the whole real-time learning and trading process. Transient effects of the initial learning while trading process can be seen in the first 2000 time steps of Figure 4 and in the distribution of differential Sharpe ratios in the lower left panel of Figure 6.

Figure 7 shows box plots summarizing test performances for ensembles of 100 experiments. In these simulations, the 10,000 data are partitioned into an initial training set consisting of the first 1000 samples and a subsequent test data set containing the last 9000 samples. The trading systems are first optimized on the training data set for 100 epochs and adapted on-line throughout the whole test data set. Each trial has different realizations of the artificial price process and different randomly chosen initial trader parameter values. We vary the transaction cost from 0.2%, 0.5% to 1%, and observe the trading frequency, cumulative profit and Sharpe ratio over the test data set. As shown, in all 100 experiments, positive Sharpe ratios are obtained.

Figures 8 and 9 compare the following three kinds of trading systems:

(1) 'Max.SR': maximizes the differential Sharpe ratio
(2) 'Max.Profit': maximizes the cumulative profit
(3) 'Min.MSE': minimizes the mean-squared forecast error.

In this comparison, the transaction costs are set to 0.5%. As shown, the 'Max.SR' and 'Max.Profit' trading systems significantly outperform the 'Min.MSE' systems. 'Max.SR' achieves slightly better mean return than 'Max.Profit' and is substantially more consistent in its performance over 100 such trials.

Portfolio management simulation

Portfolio system and data

Additional tests of the techniques are performed for trading a portfolio of securities. The portfolio trading systems are allowed to invest proportions of their wealth among three different securities with the restrictions that they must be fully invested at each time step, and that no short

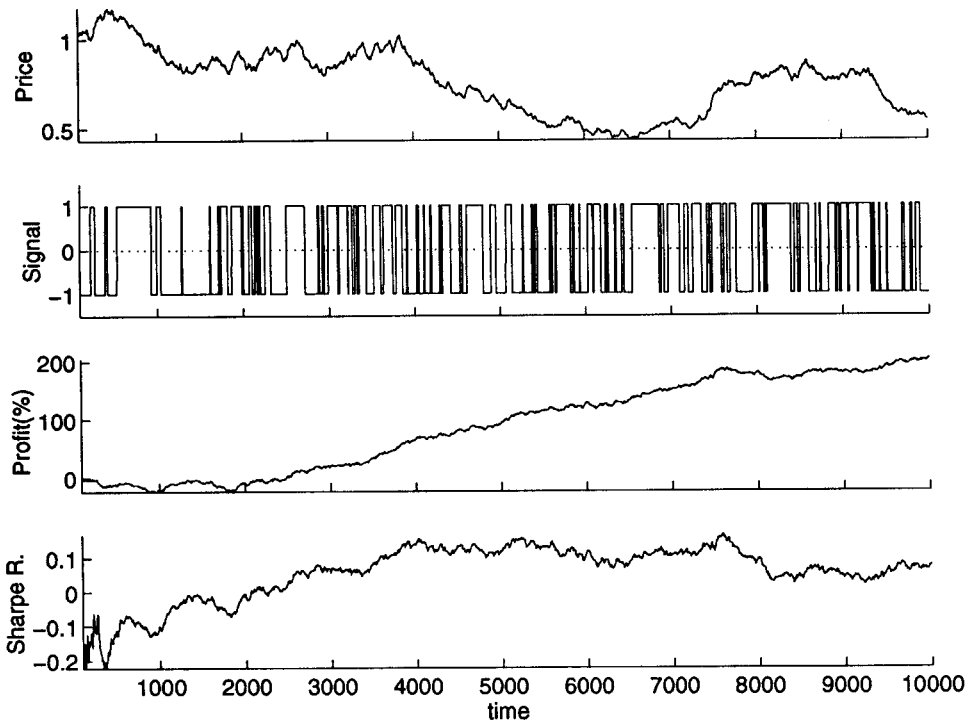

Figure 4. Artificial prices (top panel), trading signals (second panel), cumulative sums of profits (third panel) and the moving average Sharpe ratio with $\eta = 0.01$ (bottom panel). The system performs poorly while learning from scratch during the first 2000 time periods, but its performance remains good thereafter

selling is allowed. The output of a system is a set of portfolio weights $\{F_t^1, F_t^2, F_t^3\}$, with the conditions that

$$F_t^a \geqslant 0 \text{ and } \sum_{a=1}^{3} F_t^a = 1 \tag{40}$$

The recurrent state of the systems is similar to that described above.

When generating multiple price series according to the random walk model p, β, ε and ν become m dimensional vectors and α and k become $m \times m$ matrices. For these experiments we have $m = 3$, and set α to be a diagonal matrix with elements $\{0.85, 0.9, 0.95\}$ and k to be diagonal with elements $\{3, 3, 3\}$. Thus the series are independent of one another. Examples of the artificial price series are shown in the top panel of Figure 10.

Using the portfolio management system, we compare training to maximize the differential Sharpe ratio and training to maximize profits. We find for a variety of transaction costs, that on average, training to maximize the differential Sharpe ratio outperforms training to maximize profits.

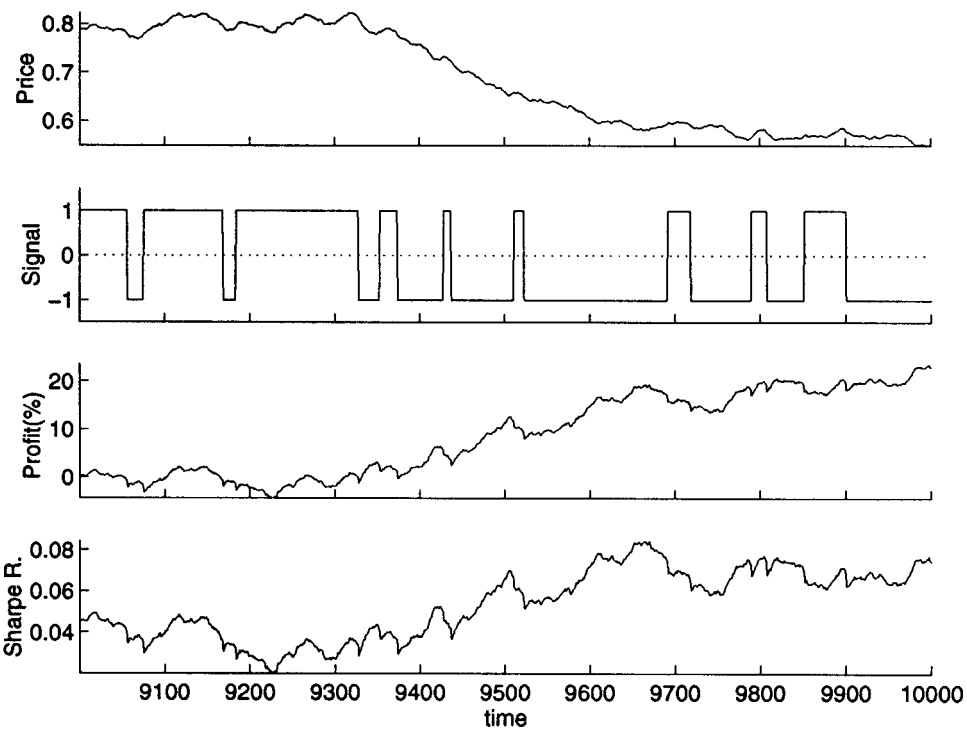

Figure 5. An expanded view of the last thousand time periods of Figure 4. The exponential moving Sharpe ratio has a forgetting time scale of $1/\eta = 100$ periods. A smaller η would smooth the fluctuations out

Simulated trading results

Figure 10 shows a section of a single simulation of the portfolio management system. The trading system starts from a random initial configuration and is then adapted to optimize the differential Sharpe ratio. The transaction costs during this simulation are set at 0.5%.

Figure 11 shows box plots summarizing test performances for ensembles of 100 experiments. In these simulations, the trading system is initialized to a random starting condition and then adapted on-line throughout the entire data set. The simulation ensembles include 10 different initializations for each of 10 different realizations of the artificial price series. We vary the transaction costs from 0.2%, 0.5% to 1%, and observe the trading frequency, cumulative profits and Sharpe ratio on the data set. The figures show that the behaviour of the portfolio management system is similar to that of the long/short trader in response to increasing transaction cost. Also, as the middle panels of Figure 10 demonstrate, the portfolio system tends to saturate the portfolio weights and take long/neutral positions in the individual securities.

Figure 12 compares training to maximize the differential Sharpe ratio ('Max.SR') and training to maximize cumulative profits ('Max.Profit'). Statistics are collected over an ensemble of 100 experiments as described previously. We see that as the transactions costs increase, the 'Max.SR' system actually outperforms the 'Max.Profit' system in terms of average final wealth. While both systems are attempting to maximize profit, it appears that in these examples, concurrently minimizing risk can have tangible effects on actual as well as risk-adjusted profits.

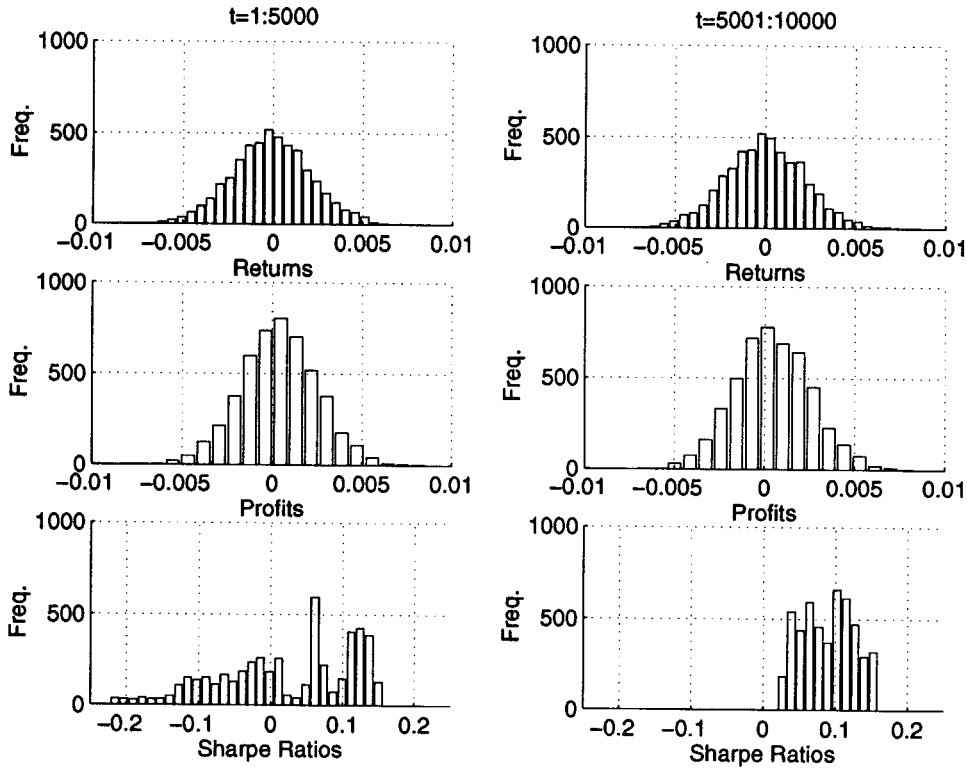

Figure 6. Histograms of the price changes (top), trading profits per time period (middle) and Sharpe ratios (bottom) for the simulation shown in Figure 4. The left column is for the first 5000 time periods, and the right column is for the last 5000 time periods. The transient effects during the first 2000 time periods for the real-time recurrent learning are evident in the lower-left graph

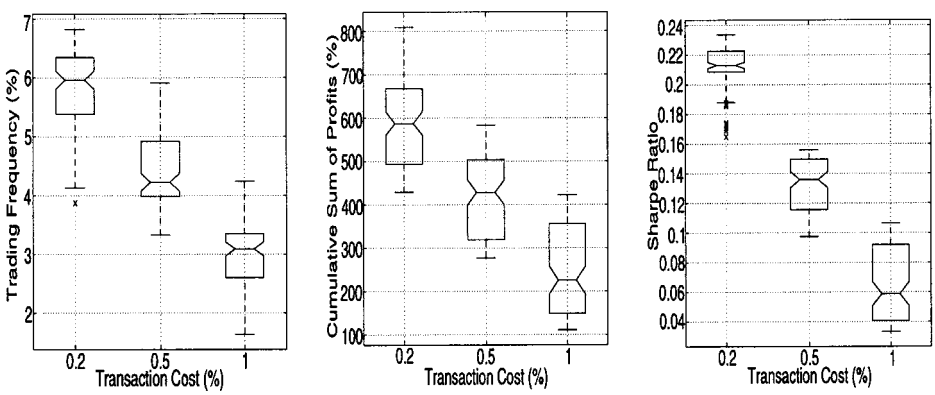

Figure 7. Boxplots of trading frequency, cumulative sums of profits and Sharpe ratios versuss transaction costs. The results are obtained over 100 trials with various realizations of artificial data and initial system parameters. Increased transaction costs reduce trading frequency, profits and Sharpe ratio, as expected. The trading frequency is the percentage of the number of time periods during which trades occur. All figures are computed on the last 9000 points in the data set

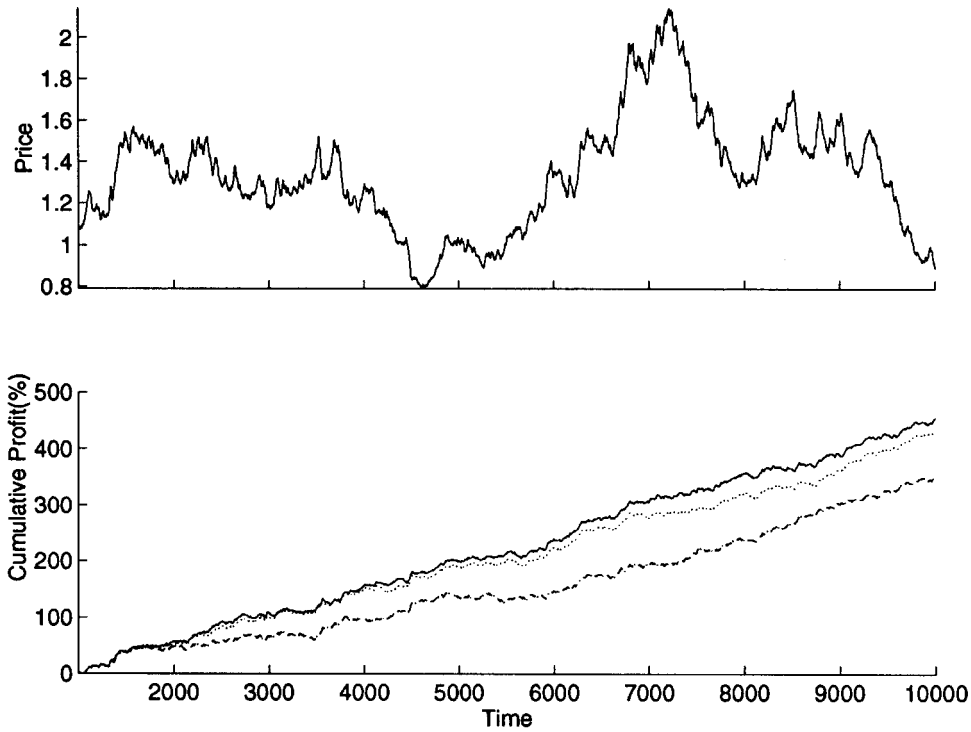

Figure 8. Comparison of the cumulative profits of three trading systems for 9000 time steps out of sample. The transaction cost is 0.5%. The price series is plotted in the upper panel. The lower panel shows the cumulative sum of median profits over 10 different trials. The solid curve is for the 'Max.SR' system, the dotted curve is for the 'Max.Profit' system and the dashed curve is for the 'Min.MSE' system

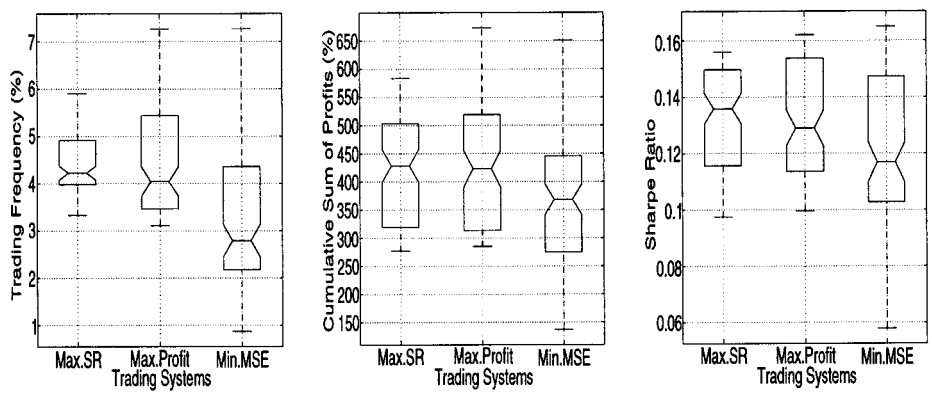

Figure 9. Boxplots of trading frequencies, profits and Sharpe ratio of three kinds of trading systems, 'Max.SR', 'Max.Profit' and 'Min.MSE'. The trading frequency is the percentage of the number of trades over the total data points. Transaction cost is 0.5%. The results are obtained over 100 trials with various realizations of artificial data and initial system parameters

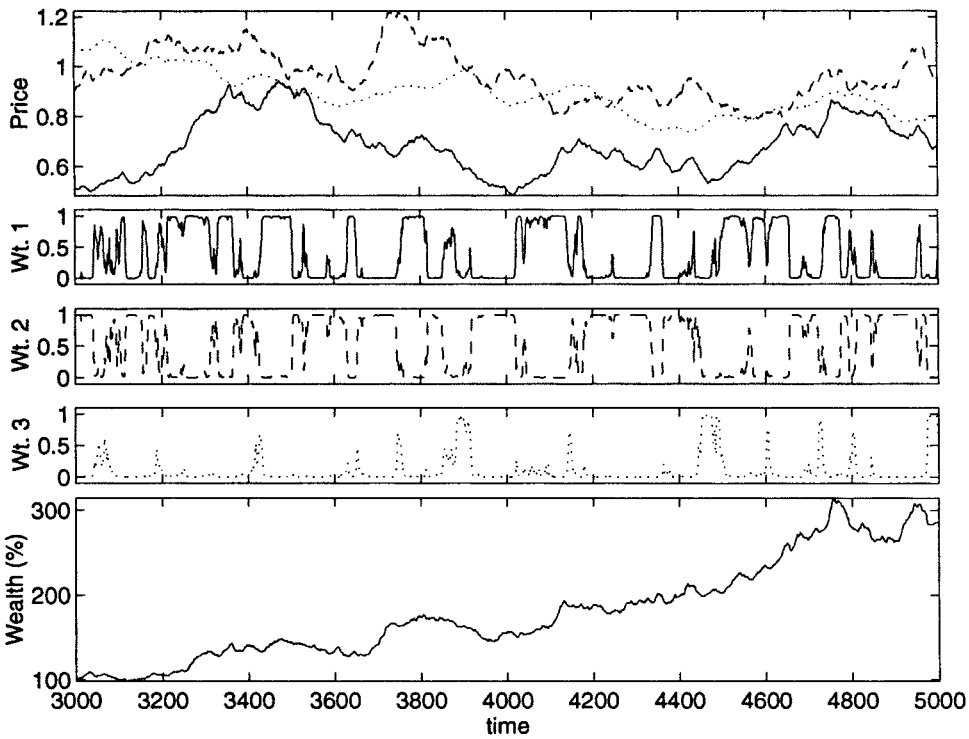

Figure 10. An expanded view of 2000 time periods from a simulation of the portfolio management system with transaction costs = 0.5%. The top panel shows the three artificial price series used in the simulation. The middle three panels show the corresponding portfolio weights chosen by the trading system at each time step. Note that the smoothest price series is also the least invested in, and that the trading system is required to be fully invested at all times. The bottom panel shows the cumulative wealth over this time period. The trading system tripled its wealth during this time period even though the price series showed almost no net gain during the period

S&P 500 / TBill asset-allocation system

Long/short asset-allocation system and data

A long/short trading system is trained on monthly S&P 500 stock index and 3-month TBill data to maximize the differential Sharpe ratio. The S&P 500 target series is the total return index computed by reinvesting dividends. The S&P 500 indices with and without dividends reinvested are shown in Figure 13 along with a 3-month treasury bill and S&P 500 dividend yields. The 84 input series include both financial and macroeconomic data. All data are obtained from Citibase, and the macroeconomic series are lagged by one month to reflect reporting delays.

A total of 45 years of monthly data are used, from January 1950 through December 1994. The first 20 years of data are used only for the initial training of the system. The test period is the 25-year period from January 1970 through December 1994. The experimental results for the 25-year test period are true *ex ante* simulated trading results.

For each year during 1970 through 1994, the system is trained on a moving window of the previous 20 years of data. For 1970, the system is initialized with random parameters. For the

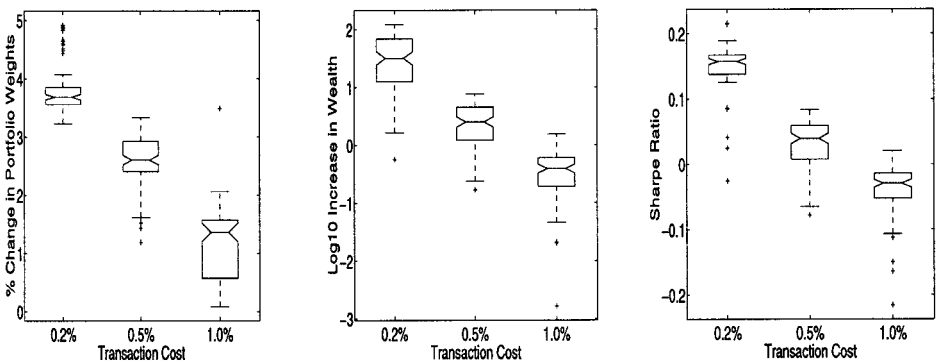

Figure 11. Boxplots of the average percentage change in the portfolio weights, cumulative profits and Sharpe ratios versus transaction costs. The results are obtained over 100 trials with various realizations of artificial data and initial system parameters. Increased transaction costs reduce the amount of change in portfolio weights, profits and Sharpe ratio, as expected. The change in portfolio weights reported here is the average of the average change of each of the three portfolio weights. All figures are computed on the last 9000 points in the data set

24 subsequent years, the previously learned parameters are used to initialize the training. In this way, the system is able to adapt to changing market and economic conditions. Within the moving training window, the first 10 years are used for stochastic optimization of system parameters, and the subsequent 10 years are used for validating early stopping of training. The networks are linear, and are regularized using quadratic weight decay during training with a regularization parameter of 0.01. For these experiments, our recurrent reinforcement learning algorithm is employed to maximize the differential Sharpe ratio.

Experimental results

Figure 14 shows results from 30 trial simulations for the S&P 500 and TBill asset allocation system during the 25-year out-of-sample test period. The transaction cost is set at 0.5%. Profits are reinvested during trading, and multiplicative profits are used when calculating the wealth. The trading systems' average performance is compared with the S&P 500 buy-and-hold strategy in the top panel. The figure also shows results for following the strategy of taking positions from a majority vote of the 30 trading systems.*

We see in Figure 14 that the trading systems go short for the S&P 500 during critical periods, such as the oil price shock of 1974, the tight money periods of the early 1980s, the market correction of 1984 and the 1987 crash. This ability to take advantage of high treasury bill rates or to avoid periods of substantial stock market loss is the major factor in the long-term success of these trading models. One exception is that the trading systems remain long during the 1991 stock market correction associated with the Gulf War.

To study the performance of the various strategies over time, Figure 15 shows the exponential moving averages of the annualized returns and Sharpe ratios attained. A time constant of $\eta = 1/24$ is used to calculate the moving averages. The trading systems achieve consistently higher annualized profits and Sharpe ratios than the S&P 500 buy-and-hold strategy.

* Combination of estimators generally outperform their component estimators (Granger and Newbold, 1986; Winkler and Makridakis, 1983; Clemen, 1989).

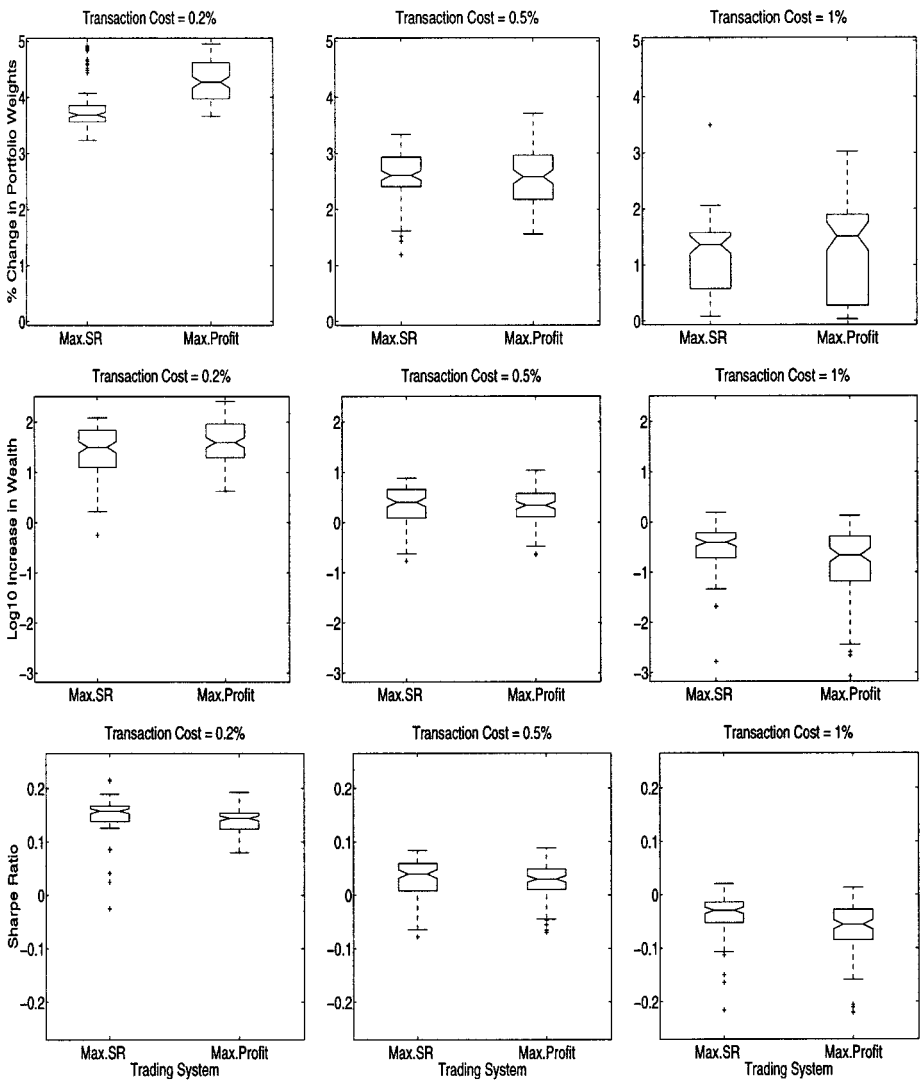

Figure 12. Boxplots of percentage change in portfolio weights, profits and Sharpe ratios of the two trading systems, 'Max.SR' and 'Max.Profit'. The change in portfolio weights reported here is the average of the average change of each of the three portfolio weights. Transaction costs are 0.2%, 0.5% and 1%. The results are obtained over 100 trials with various realizations of artificial data and initial system parameters

Figure 16 shows box plots summarizing the test performance for the full 25-year test period of the trading system over 30 trials with various realizations of the initial system parameters. The notches in the box plots indicate robust estimates of the 95% confidence intervals on the hypothesis that the median is equal to the performance of the buy-and-hold strategy. The horizontal lines show the performance of the average, voting and buy-and-hold strategies for the same test period. As can be seen in Figure 16, following the voting strategy results in higher profit

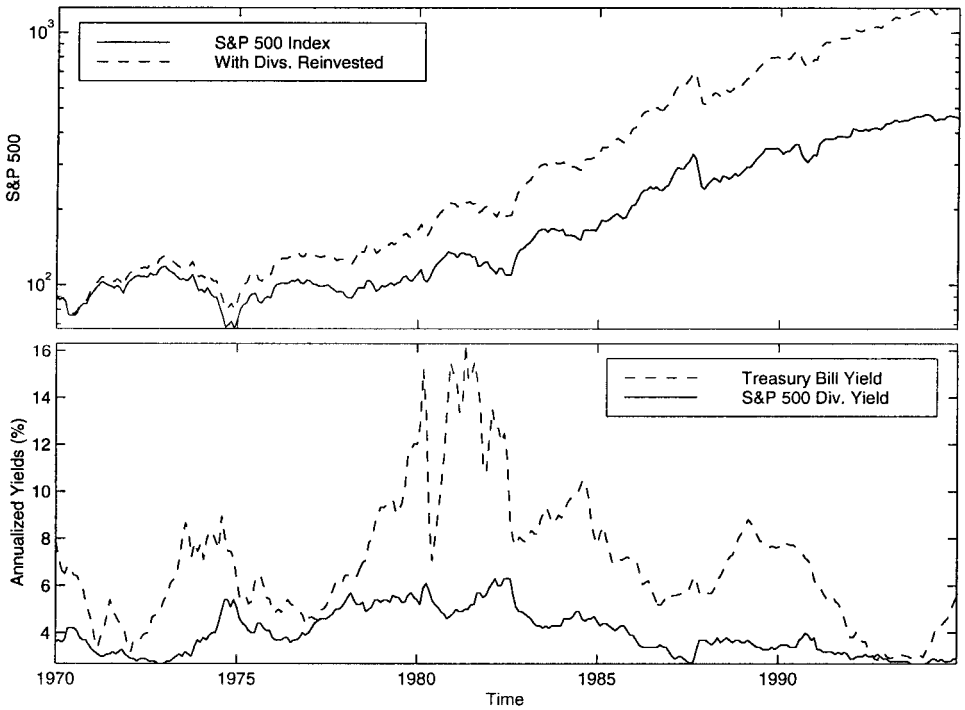

Figure 13. Time series that influence the return attainable by the S&P 500/TBill asset allocation system. The top panel shows the S&P 500 series with and without dividends reinvested. The bottom panel shows the annualized monthly Treasury Bill and S&P 500 dividend yields

than the average performance of the individual traders, but a lower Sharpe ratio value. The annualized monthly Sharpe ratios of the buy-and-hold strategy, the average strategy and the voting strategy are 0.34, 0.84 and 0.83 respectively. The Sharpe ratios calculated here are for the excess returns of the strategies over the 3-month treasury bill rate.

From these results we find the trading system outperforms the buy-and-hold strategy, as measured by both accumulated wealth and Sharpe ratio, and is less risky, as measured by maximum drawdown. These differences are statistically significant and support the proposition that there is predictability in the US stock and treasury bill markets during the 25-year period 1970 through 1994.

CONCLUSIONS

We have proposed and tested a methodology for training trading systems and portfolios by optimizing objective functions that directly measure trading and investment performance. The performance functions that we optimize include profit and wealth, economic utility functions, the Sharpe ratio, our proposed *differential Sharpe ratio*, and other performance ratios.

Rather than basing a trading system on forecasts or training via a supervised learning algorithm using labelled trading data, we train our systems using reinforcement learning algorithms. The trading and portfolio management systems require prior decisions as input in order to

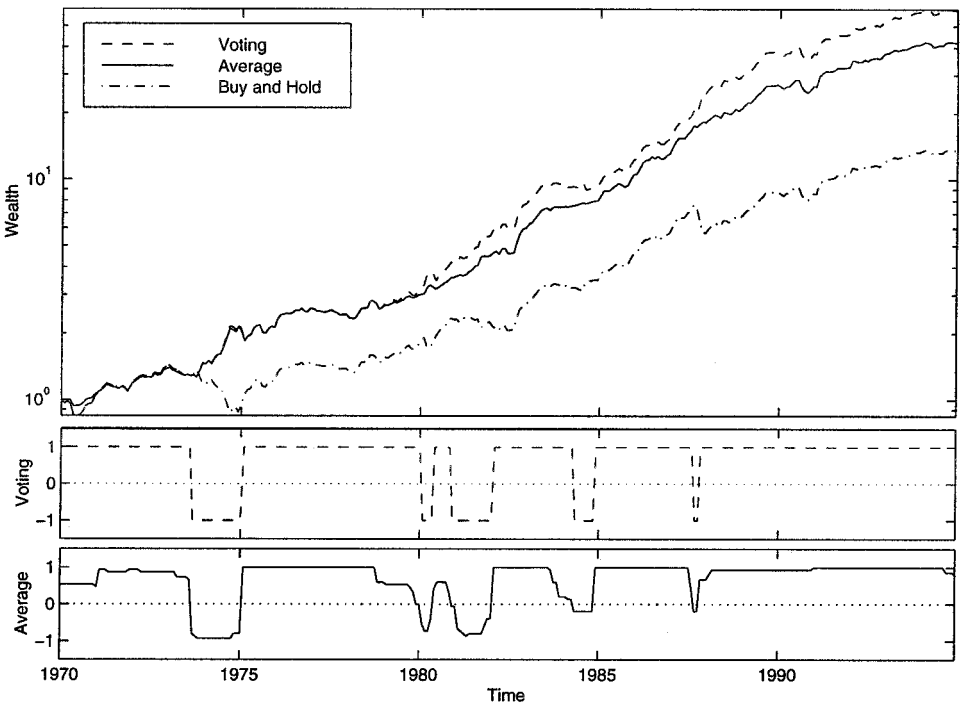

Figure 14. Equity curves and trading signals for the monthly S&P 500/Tbill asset allocation system during the test period for 30 trial simulations with various realizations of the initial randomly-selected system parameters. The top panel shows the cumulative wealth for the average of the 30 trading systems (solid curve), the wealth from following the voting strategy (dashed curve) and the wealth from the buy and hold strategy (dots and dashes). The second panel shows the voting trading position over 30 trials. The third panel shows the average trading position taken by the 30 traders

properly take into account the effects of transactions costs, market impact and taxes. This temporal dependence on system state requires the use of recurrent learning algorithms. Hence, we train our systems using reinforcement versions of standard recurrent learning algorithms.

Reinforcement learning algorithms find approximate solutions to stochastic dynamic programming problems and are able to do so on-line. Although it is possible to train the systems off-line using batch learning, we favour on-line reinforcement learning, as it is more efficient computationally. The learning algorithms we use are thus stochastic optimization methods. We utilize a simple but unique recurrent reinforcement learning (RRL) algorithm based on real-time recurrent learning (RTRL) that maximizes immediate rewards in an on-line mode.

To enable on-line optimization of the Sharpe ratio, we have developed the differential Sharpe ratio. This offers certain advantages relative to the standard Sharpe ratio in terms of ease of computation and interpretability. We compare the form of the differential Sharpe ratio to the Bernoulli and power law economic utility functions, and point out that the Sharpe ratio behaves like an adaptive (non-stationary) utility function, in that it depends upon past experience.

The empirical results presented demonstrate the efficacy of several of our methods. For a long/short trader of a single artificial price series, we find that maximizing the differential Sharpe ratio yields more consistent results than maximizing profits, and that both methods outperform a

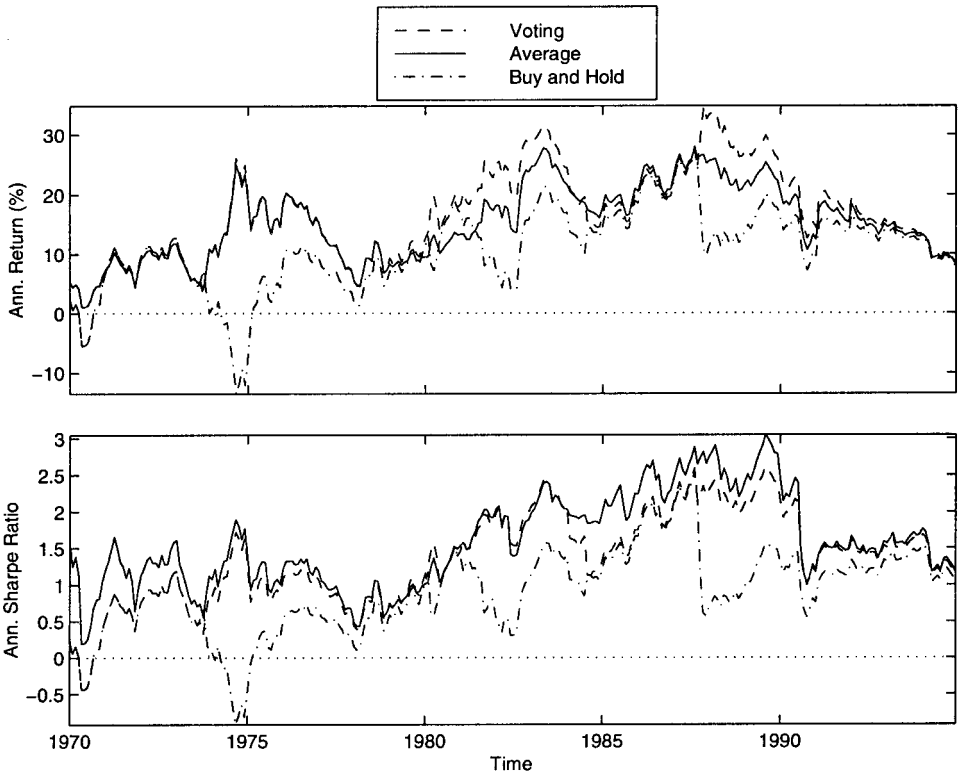

Figure 15. Performance results of 30 trial simulations for the monthly S&P 500 data during the test period. The top panel shows the exponential moving averages of annualized monthly returns for the voting trading system (dashes), average trading system (solid) and the buy and hold strategy (dots and dashes). The bottom panel shows the exponential moving average Sharpe ratios for the same strategies. A forgetting factor $\eta = 1/24$ is used to calculate the moving averages

trading system based on forecasts that minimize MSE. For portfolios of three artificial price series, we find that ensembles of traders trained to maximize the differential Sharpe ratio perform better than those trained to maximize profit. The Sharpe ratio traders have smaller variations in portfolio weights and achieve higher risk-adjusted returns.

Our simulation results for the monthly S&P 500/TBill asset allocation system demonstrate the existence of predictable structure in US stock and treasury bill market returns for the twenty-five year period 1970 through 1994. The accumulated profits, Sharpe ratios, and maximum drawdowns for an ensemble of 30 trading systems are substantially better than for a buy-and-hold strategy with dividends reinvested. These results are statistically significant.

The original presentation of our approach appears in Moody and Wu (1996, 1997). The results for the S&P 500/TBill asset allocation system were first presented in Moody *et al.* (1998), along with a comparison of Q-Learning (Watkins, 1989; Watkins and Dayan, 1992) to the RRL algorithm presented here. In that paper, we show that our RRL algorithm provides more stable results and higher profits and Sharpe ratios than does the Q-Learning algorithm for the 25-year out-of-sample period for the S&P 500/TBill asset-allocation system.

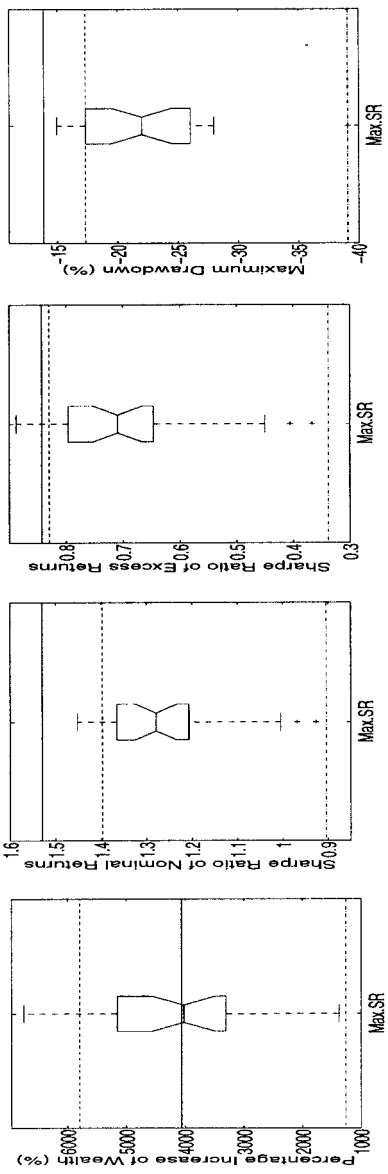

Figure 16. Performance statistics for the S&P 500/TBill asset allocation system. The boxplots show the percentage increase of wealth, annualized monthly Sharpe ratios of nominal returns, annualized monthly Sharpe ratios of excess returns over the three month treasury bill rate and maximum drawdown of the 'Max.SR' trading system. The results are for the monthly S&P 500 series (with dividends reinvested) and monthly TBill series for the test period of January 1970 through December 1994. The results are obtained over 30 trials with various realizations of the initial randomly-selected system parameters. The horizontal lines show the performance of the average strategy (solid line), the voting strategy (dashed line) and the S&P 500 buy and hold strategy (dots and dashes) for the same test period. The notches in the box plots indicate robust estimates of the 95% confidence intervals on the hypothesis that the median is equal to the performance of the buy and hold strategy. The differences in performance between the ensemble of trading systems and the buy and hold strategy are statistically significant, and demonstrate the presence of predictable structure in the U.S. stock and treasury bill markets during the 1970 through 1994 test period

ACKNOWLEDGEMENTS

We thank Allan Timmermann and the anonymous referees for helpful comments. We gratefully acknowledge support for this work from Nonlinear Prediction Systems and from DARPA under contract DAAH01-96-C-R026.

REFERENCES

Barto, A., Sutton, R. and Anderson, C., 'Neuronlike elements that can solve difficult learning control problems', *IEEE Transactions on Systems, Man, and Cybernetics*, **13** (1983), 835–846. Reprinted in J. A. Anderson and E. Rosenfeld (eds), *Neurocomputing: Foundations of Research*, Cambridge, MA: MIT Press, 1988.

Bengio, Y., 'Training a neural network with a financial criterion rather than a prediction criterion', in Y. Abu-Mostafa, A. N. Refenes and A. Weigend (eds), *Decision Technology for Financial Engineering*, London: World Scientific, 1997.

Bernoulli, D., 'Exposition of a new theory on the measurement of risk', *Econometrica*, **32** (1954), 23–26.

Bertsekas, D. P., *Dynamic Programming and Optimal Control*, Belmont, MA: Athena Scientific, 1995.

Clemen, R. T., 'Combining forecasts: A review and annotated bibliography', *International Journal of Forecasting*, **5** (1989), 559–583.

Granger, C. and Newbold, P., *Forecasting Economic Time Series*, New York: Academic Press, 1986.

Grinold, R. C. and Kahn, R. N., *Active Portfolio Management: Quantitative Theory and Applications*, New York: McGraw-Hill, 1995.

Kang, J., Choey, M. and Weigend, A., 'Nonlinear trading models and asset allocation based on Sharpe ratio', in Y. Abu-Mostafa, A. N. Refenes and A. Weigend (eds), *Decision Technology for Financial Engineering*, London: World Scientific, 1997.

Markowitz, H., *Portfolio Selection: Efficient Diversification of Investments*, New York: Wiley, 1959.

Moody, J. and Wu, L., 'Optimization of trading systems and portfolios, neural networks in the capital markets', (NNCM*96) Conference Record, Caltech, Pasadena, 1996.

Moody, J. and Wu, L., 'Optimization of trading systems and portfolios', in Y. Abu-Mostafa, A. N. Refenes and A. Weigend (eds), *Decision Technologies for Financial Engineering*, London: World Scientific, 1997, pp. 23–35. This is a slightly revised version of the original paper that appeared in the NNCM*96 Conference Record, published by Caltech, 1996.

Moody, J., Saffell, M., Liao, Y. and Wu, L., 'Reinforcement learning for trading systems and portfolios: Immediate vs future rewards', in A. N. Refenes, N. Burgess and J. Moody (eds), *Decision Technologies for Financial Engineering*, Amsterdam: Kluwer, 1998. This volume is the proceedings for the 1997 Computational Finance conference held at the London Business School.

Narendra, K. S. and Parthasarathy, K., 'Identification and control of dynamical systems using neural networks', *IEEE Transactions on Neural Networks*, **1**(1) (1990), 4–27.

Nawrocki, D., 'Optimal algorithms and lower partial moment: *Ex post* results', *Applied Economics*, **23** (1991), 465–470.

Nawrocki, D., 'The characteristics of portfolios selected by n-degree lower partial moment', *International Review of Financial Analysis*, **1** (1992), 195–209.

Neuneier, R., 'Optimal asset allocation using adaptive dynamic programming', in D. Touretzky, M. Mozer and M. Hasselmo (eds), *Advances in Neural Information Processing Systems 8*, Cambridge, MA: MIT Press, 1996.

Rumelhart, D., Hinton, G. and Williams, R., 'Learning internal representations by error propagation', in D. Rumelhart and J. McClelland (eds), *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, Cambridge, MA: MIT Press, 1986, chapter 8, pp. 319–362.

Samuelson, P. A., 'Asset allocation could be dangerous to your health', *The Journal of Portfolio Management*, (1990), 5–8.

Satchell, S. and Timmermann, A., 'An assessment of the economic value of non-linear foreign exchange rate forecasts', *Journal of Forecasting*, **14** (1995), 477–497.

Sharpe, W. F., 'Mutual fund performance', *Journal of Business*, (1966), 119–138.

Sharpe, W. F., 'The Sharpe ratio—properly used, it can improve investment management', *The Journal of Portfolio Management*, (1994), 49–58.
Sortino, F. and Forsey, H., 'On the use and misuse of downside risk', *The Journal of Portfolio Management*, **22** (1996), 35 + .
Sortino, F. and Vandermeer, R., 'Downside risk—capturing what's at stake in investment situations', *The Journal of Portfolio Management*, **17** (1991), 27–31.
Sutton, R. S., 'Learning to predict by the methods of temporal differences', *Machine Learning*, **3** (1988), 9–44.
Sutton, R. S. and Barto, A. G., *An Introduction to Reinforcement Learning*, Cambridge, MA: MIT Press, 1997.
Timmermann, A. and Pesaran, H., 'Predictability of stock returns: Robustness and economic significance', *Journal of Finance*, **50** (1995), 1201–1228.
Treynor, J. and Black, F., 'How to use security analysis to improve portfolio selection', *Journal of Business*, (1973), 68–86.
Watkins, C. J., *Learning with Delayed Rewards*, PhD thesis, Cambridge University, Psychology Department, 1989.
Watkins, C. J. and Dayan, P., 'Technical note: Q-Learning', *Machine Learning*, **8**(3) (1992), 4.
Werbos, P., 'Back-propagation through time: What it does and how to do it', *IEEE Proceedings*, **78**(10) (1990), 1550–1560.
White, H., Personal communication. Unpublished, 1996.
Widrow, B. and Hoff, M. E., 'Adaptive switching circuits', in *IRE WESCON Convention Record*, (1960), pp. 96–104.
Williams, R. J. and Zipser, D., 'A learning algorithm for continually running fully recurrent neural networks', *Neural Computation*, **1** (1989), 270–280.
Winkler, R. L. and Makridakis, S., 'The combination of forecasts', *Journal of Royal Statistical Society*, (1983), (146).

Authors' biographies:

John Moody is the Director of the Computational Finance Program and a Professor of Computer Science and Electrical Engineering at Oregon Graduate Institute of Science and Technology. He is also the Founder and President of Nonlinear Prediction Systems, a company specializing in the development of forecasting and trading systems. Moody is a past General Chair and Program Chair of the Neural Information Processing Systems (NIPS) conference and serves on the Organizing Committees of the Computational Finance and Forecasting Financial Markets conferences. He received his BA in Physics from the University of Chicago (1979) and his MA and PhD in Theoretical Physics from Princeton University (1981 and 1984).

Lizhong Wu is a Principal Project Scientist at Oregon Graduate Institute and a Senior Consulting Scientist at Nonlinear Prediction Systems. He received his BSc and MSc in Electrical Engineering from South China University of Science and Technology in 1983 and 1986, and his PhD in Information Engineering from Cambridge University in 1992. He was a postdoctoral research fellow at Engineering Department of Cambridge University in 1992–1993.

Yuansong Liao is a PhD candidate in the Department of Computer Science and Engineering at the Oregon Graduate Institute. She received her BSc in Computer Science and Engineering from Huazhong University of Science and Technology, China, in 1989, and her MPhil in Computer Speech and Language Processing from the University of Cambridge in 1992.

Matthew Saffell is a PhD candidate in the Computer Science and Engineering Department at the Oregon Graduate Institute and a Consulting Scientist at Nonlinear Prediction Systems. He received his BSc in Computer Science and Engineering from Le Tourneau University in 1992, and his MSc in Computer Science from the University of Tennessee in 1994.

Authors' address:
John Moody, **Lizhong Wu**, **Yuansong Liao** and **Matthew Saffell**, Oregon Graduate Institute of Science and Technology, Department of Computer Science and Engineering, PO Box 91000, Portland, OR 97291-1000, USA.

