Knowledge-Based Systems 79 (2015) 27–35

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

An evolutionary trend reversion model for stock trading rule discovery

Xiangzhou Zhang [a], Yong Hu [b,*], Kang Xie [a], Weiguo Zhang [c], Lijun Su [d], Mei Liu [e]

[a] School of Business, Sun Yat-sen University, No. 135, Xingang Xi Road, Guangzhou 510275, PR China
[b] Institute of Business Intelligence and Knowledge Discovery, Guangdong University of Foreign Studies, Sun Yat-sen University, Guangzhou 510006, PR China
[c] School of Business Administration, South China University of Technology, Wushan Road, Tianhe District, Guangzhou 510641, PR China
[d] School of Management, Guangdong University of Foreign Studies, Higher Education Mega Center, Guangzhou 510006, PR China
[e] University of Kansas Medical Center, Kansas City, KS 66160, USA

ARTICLE INFO

Article history:
Available online 17 August 2014

Keywords:
Quantitative investment
Trading rule discovery
Trend reversion
eXtended Classifier System (XCS)
XCS with *learn* mode (XCSL)

ABSTRACT

Quantitative investment (QI) is certainly a hot topic in big data analysis. For knowledge discovery in huge, complex and nonlinear stock market data, the eXtended Classifier Systems (XCS) is quite suitable because of the excellent learning and explicit expression abilities derived from its intrinsic techniques that include classification rule mining, evolutionary learning and reinforcement learning. This paper presents an Evolutionary Trend Reversion Model (eTrendRev), which is based on the proposed XCS with *learn* mode (XCSL) and trend-reversion strategy. The eTrendRev is highlighted in three aspects: (1) the explicit rules generated by XCSL are more understandable than black-box models, such as neural networks, thus can provide justifiable knowledge to guide trading; (2) the original *pure explore* mode of XCS is substituted by the proposed *learn* mode, which is shown in this study to perform better and is more stable; (3) a variety of trend-reversion strategies are integrated and made dynamic through evolutionary learning. For model evaluation, experiments were carried out on the historical data of the Shanghai Composite Index and the NASDAQ Composite Index, and back-testing results indicate that eTrendRev can produce higher return with lower risk and recognize significant market turning points in a timely fashion. This study also confirms the profitability of using sole trend-reversion indicators in machine learning-based QI model.

1. Introduction

Quantitative investment (QI) is an important research topic for stock trading in the big data era [25]. Big data in stock market are of high volume, high velocity, high variety and high variability (4V), including heterogeneous data sources—real-time (or online) and offline transaction data of different time granularities, as well as (textual) news, financial statements and macroeconomic statistics released by governments, institutions, social media and listed companies. Great challenges exist for knowledge discovery in stock trading/transaction data because they are generally known to be non-linear, noisy, and of high dimension [4,5]. Different from the traditional expert/individual experience-based trading strategies, QI models adopt systemic and quantitative methods, such as statistics, artificial intelligence, machine learning (ML) and data mining techniques. Owing to these techniques of big data analysis, QI models are effective in discovering knowledge from abundant homogeneous and/or heterogeneous stock market data, and thus have attracted increasing interest.

However, "black-box" QI models, such as those based on neural networks [22], lack interpretability and cannot provide understandable knowledge about the inter-actions between trading strategy indicators (i.e., model inputs) and price movement (i.e., model output). Investors often feel uncomfortable and insecure about making stock trading decisions with a "black-box" model [10]. To transform this negative perception, researchers have invested a lot of effort in finding comprehensible stock trading rules, especially those correspond with classical investment strategies, such as trend following and reversion [11,13]. Therefore, both "black-box" and "comprehensibility" issues must be considered.

To address the "black-box" issue, the eXtended Classifier Systems (XCS) is quite suitable. XCS has excellent learning and explicit expression abilities, because it delicately combines three representative ML technologies, including classification rule mining (i.e., non-black-box), evolutionary learning and reinforcement learning. In stock market prediction/trading, XCS-based QI models were

* Corresponding author. Tel.: +86 20 39328876; fax: +1 973 596 5777.
E-mail addresses: zhxzhou@mail2.sysu.edu.cn (X. Zhang), henryhu200211@163.com (Y. Hu), mnsxk@mail.sysu.edu.cn (K. Xie), wgzhang@scut.edu.cn (W. Zhang), onnowa@163.com (L. Su), meiliu@kumc.edu (M. Liu).

http://dx.doi.org/10.1016/j.knosys.2014.08.010
0950-7051/

demonstrated to be more profitable compared to the random and buy-and-hold strategies [9,18]. Advantages of XCS include: (1) XCS can properly learn explicit rules from noisy, complex and non-linear environment that continuously changes, (2) XCS can make real-time and accurate learning and response, and (3) XCS can adjust itself to strengthen its inward knowledge step by step [33]. However, the original *pure explore* mode of XCS is not optimal for stock trading because this random-exploring strategy cannot adequately use historical information. For instance, when previous information regarding stock price movement is available, it is unnecessary to "randomly guess" the best action (i.e., buy or sell a stock). Thus, it is imperative to design a new strategy/mode for "learning from history".

To address the "comprehensibility" issue, it is necessary to integrate classical investment strategies into ML-based QI model. Trend-following QI models have been widely examined as simple but effective investment tools [10,12,11]). Trend-reversion ML-based QI model, however, has attracted less attention. Actually, the premise of trend-reversion strategy is the mean reversion/reversal theory of stock price movement, which has been statistically verified in numerous studies [3,11]. Our paper explores the profitability of using sole trend-reversion indicators in a ML-based QI model.

This paper proposes a novel XCS with *learn* mode (XCSL) and establishes an Evolutionary Trend Reversion Model (eTrendRev) that integrates trend-reversion strategy with XCSL. The eTrendRev is highlighted in three aspects: (1) XCSL is able to generate explicit rules that are more understandable than black-box models, such as neural networks, and can provide justifiable knowledge to guide stock trading; (2) the original *pure explore* mode of XCS is substituted by the proposed *learn* mode, which can make full use of historical information and lead to more stable and better performance; (3) a variety of trend-reversion strategies are integrated and made dynamic through the evolutionary learning framework of XCSL.

To evaluate eTrendRev, experiments were carried out on the historical data of the Shanghai Composite (SH) Index of the Chinese stock market and the NASDAQ Composite (NASDAQ) Index of the US stock market. In addition, a brief analysis of buy/sell signals on market turning points is provided.

The rest of the paper is organized as follows. A brief review of related works is presented in Section 2. The architecture and internal logic of eTrendRev are described in Section 3. Section 4 presents experiment results and analysis. Finally, a summary and conclusion is provided in Section 5.

2. Literature review

2.1. Quantitative stock trading strategy

With respect to stock trading, a quantitative strategy differs from a discretionary strategy in that quantitative trading can reduce the arbitrariness of discretionary trading by utilizing computerized and systematic methods to eliminate subjective decisions driven by emotion, indiscipline, passion, greed, and fear. Generally, quantitative models for stock selection and/or timing can be grouped into two categories: (1) theory-driven and (2) data-driven [26]. They have the same objective of "forecasting" stock price movement based on historical observational data. Specifically, the former first speculates generalized investment/financial theories and then rigorously tests them, such as in ([3,16], while the later pays less attention to theory explanation and directly adopts data analysis techniques (e.g. statistics, artificial intelligence, ML and/or data mining) to uncover hidden patterns, such as in [6,22,36]. The present study belongs to the latter data-driven paradigm.

As for trend reversion, most studies follow the theory-driven paradigm and focus on statistical verification of mean reversion/reversal effect [3,11]. For example, Cunado et al. [11] tested whether mean reversion in stock prices presents a different behavior in US bull and bear markets and found that different episodes of mean reversion mainly correspond to bull market periods. For another example, Balvers and Wu [3] evaluated the pure momentum, pure mean-reversion and the joint momentum and mean-reversion strategies across 18 countries with well-developed equity markets.

However, few studies have investigated trend reversion under the data-driven paradigm, for example, examining the effectiveness of sole trend-reversion indictors in ML-based QI models. The above theory-driven studies [3,11] construct their trend-reversion portfolio solely based on stock return information, whereas a data-driven ML-based model can incorporate numerous well-verified trend-reversion indicators that are available in literature and practice.

2.2. Trend-reversion indicators and rules

In the common practice of technical analysis, trend-reversion indicators, or price oscillators, are used to explain where the current price lies within the recent range and generate buy signal when price has fallen too low and sell signal when price has risen too high [20]. The underlying assumption of trend reversion is the mean reversion theory, i.e., once price moves too far from average, a reversal is eminent. Popular trend-reversion indicators include the Relative Strength Index (RSI), Stochastic Oscillator (K&D), and Moving Average Convergence/Divergence (MACD).

2.2.1. RSI

RSI is perhaps the most popular over-bought and over-sold indicator used by investors for technical analysis [20,21]. RSI compares the average price changes of up days with average price changes of down days in an attempt to determine over-bought and over-sold conditions of an asset. RSI is defined as follows:

$$\mathrm{RSI}(n) = 100 - \frac{100}{1 + \frac{U}{D}},$$

U denotes the average price changes of up days over a n-day period, and D denotes the average price changes of down days over the same period.

Price always gets to a peak when RSI rises above 70, while bottom occurs when RSI falls below 30. Therefore, RSI can be used as a trend-reversion indicator. When RSI is below 30, a buy signal is generated once RSI rises back above 30. Conversely, when RSI is above 70, a sell signal is generated once RSI enters on a cross below 70.

2.2.2. K&D

The stochastic (fast $\%K$ and fast $\%D$) indicator is another popular momentum oscillator, which measures the strength of price trend and indicates potential short-term market over-bought and over-sold levels [19]. It consists of two variables, fast $\%K$ and fast $\%D$, defined at each time t as:

$$\%K(m)_t = 100\frac{P_t - P_t^{min}(m)}{P_t^{max}(m) - P_t^{min}(m)},$$

where $P_t^{min}(m)$ and $P_t^{max}(m)$ are the minimum and maximum closing prices over a m-day period, respectively, and

$$\%D(n)_t = \sum_{i=1}^{n} \frac{\%K(m)_{t-i}}{n},$$

is the n ($n < m$) days moving average of $\%K$.

Buy signal is generated when the %*K* rises above 20 and is accompanied by a cross above the %*D*, while sell signal is generated when the %*K* falls below 80 and is accompanied by a cross below the %*D*.

2.2.3. MACD

The *MACD* is an oscillator calculated by taking the difference between two exponential averages (a short-term and a long-term exponentially weighted moving average) [19]. First, the exponential moving average (*EMA*) of the closing price over a *n*-day period is expressed as

$$EMA(n)_t = \frac{1}{n}P_t + \left(1 - \frac{1}{n}\right)EMA(n)_{t-1},$$

where P_t is the closing price at time *t*. Then the MACD can be defined as:

$$MACD(n_1, n_2)_t = EMA(n_1)_t - EMA(n_2)_t,$$

where $n_1 < n_2$. Finally, the *MACD* signal (*MACDS*) can be define as:

$$\begin{aligned} MACDS(n_1, n_2, n_3)_t &= n_3\, day\, EMA\, of\, MACD(n_1, n_2)_t \\ &= \frac{1}{n_3}MACD(n_1, n_2)_t + \left(1 - \frac{1}{n_3}\right)MACD(n_1, n_2)_{t-1}. \end{aligned}$$

The short-term average moves up/down faster than the long-term average, and thus their crossovers always are significant price turning points [20]. As the *MACD* rises, stalls, and then crosses below the *MACDS*, it indicates that the trend is exhausted and a sell signal should be generated. Conversely, if the *MACD* falls, trades flat, and then rises above the *MACDS*, it implies that over-sold exists and a buy signal should be generated.

2.3. XCS-based stock prediction/trading model

XCS was first proposed in Wilson [35], which originated from Holland's canonical framework of learning classifier systems [17]. XCS simulates the interaction with an environment via detectors for sensing inputs and effectors for taking actions. In addition, the environment provides a scalar reward/reinforcement, through which reinforcement learning is conducted. And genetic algorithm (GA) is adopted for evolutionary learning to produce new classifiers (i.e., action rules).

Recently, XCS has been applied in studies on stock prediction and trading, such as [9,18,33]. For example, Hsu et al. [18] proposed a dynamic learning and adjustable XCS inter-market arbitrage trading model, which yielded sufficient accuracy and profitability and lower risks compared to random trading strategies. Chen and Chen [9] valued the outstanding rule discovery capability of XCS and adopted XCS to conduct a three-phase knowledge extraction. The extracted rules were then used for financial investment prediction, resulting in a better performance than fuzzy rules and decision tree.

2.4. Limitation and improvement of XCS

Despite the advantages of XCS, its quite complex mechanism and components lead to a variety of limitations. For example, XCS is unable to perform efficiently in non-Markov environments[1] due to the lack of a messaging mechanism [35], which has been the source of inspiration for memory-based XCS, such as XCSM and XCSMH [23,24]. In recent years, increasing effort has been devoted to improve XCS.

Troć and Unold [31] addressed the parameter-setting drawback of XCS and proposed a novel approach for self-adaptation of the learning rate parameter (β_p) in noisy and dynamic environments.

Zang et al. [37] claimed that the traditional XCS has shortcoming in solving large multi-step problems with delayed rewards, because the discounted reinforcement learning method used in XCS, *Q*-learning, limits the length of action chains. Thus, they employed an undiscounted reinforcement learning technique, *R*-Learning, to replace *Q*-learning and their XCSAR (where "AR" stands for "average reward") can rapidly solve large maze problems (e.g., *Woods14*).

Shariat Panahi et al. [28] pointed out that the equal-treatment policy of XCS for both correct- and incorrect-responding classifiers often leads to inefficient rule sets. To address this problem, they proposed the Success Rate XCS (SRXCS) which features an experience-evaluation mechanism that can affect a classifier's chance to survive (and then reproduce) and to deletion.

To the best of our knowledge, there is no related study that addresses the shortcomings of the *pure explore* mode of XCS (especially in stock trading) and/or develops a similar mechanism as the proposed *learn* mode. In addition, for better and independent validation of the proposed *learn* mode, this paper chooses the original XCS rather than XCS variants for performance comparison.

3. eTrendRev

3.1. Description

An overall picture of eTrendRev is given in Fig. 1. The eTrendRev takes XCSL as its core, and is designed for selecting the optimal actions in a stock market environment. Following the representative presentation in [35], Fig. 1 mainly depicts the XCS mechanism, including its data sources, data flow and process flow.

With respect to data sources, input data of eTrendRev can be offline and online daily stock trading data, including the open, high, low, closing prices, etc. These data are first used to calculate the trend-reversion and other status indicators that represent the current states of environment. Besides, these data are also used to calculate environment reward of each executed action.

The internal XCS sustains a set of classifiers (i.e., trading rules), and this classifier population is labeled by [*P*] in Fig. 1. Each classifier consists of a condition on the left and an action on the right, in the form of "*condition*:*action*" (or "IF *condition* THEN *action*"). The action part is a symbolic representation of an action that can be executed in the environment—"0" for sell signal and "1" for buy signal, while the condition part, a string of *N* bits encoded over the ternary $\{0, 1, \#\}$, is used to match the environment states. A condition bit is set "#" if it matches any value of $\{0, 1\}$. In addition, each classifier is associated with three important parameters, including prediction, prediction error, and fitness parameters, symbolized by *p*, ε, and *F*, respectively.

Given an input (i.e., environment states), a match set [*M*] is formed out of the population [*P*]. A prediction value $P(a_i)$ for each action a_i presented in [*M*] is formed, which indicates the system's "guess" of the expected payoff/reward to be received if a_i is executed. These $P(a_i)$ values are placed in a prediction array, and an action, a^*, is selected as the best action according to specific action-selection regimes (e.g., selecting the action with maximum prediction). Classifiers of action a^* forms the action set [*A*]. When the action is executed and the environment reward is received, the system updates the *p*, ε, and *F* values of the classifiers in the action set [*A*] in a reinforcement learning way [35].

Moreover, genetic algorithm (GA) is applied on the match set [*M*] to discover new classifiers (i.e., classifier reproduction). The XCS not only performs common GA operations including selection, crossover, and mutation, but also contains a covering mechanism

---

[1] An environment has the Markov property if the agent's immediate perception contains all necessary information for choosing the best action in every step; otherwise, it is non-Markov [14].

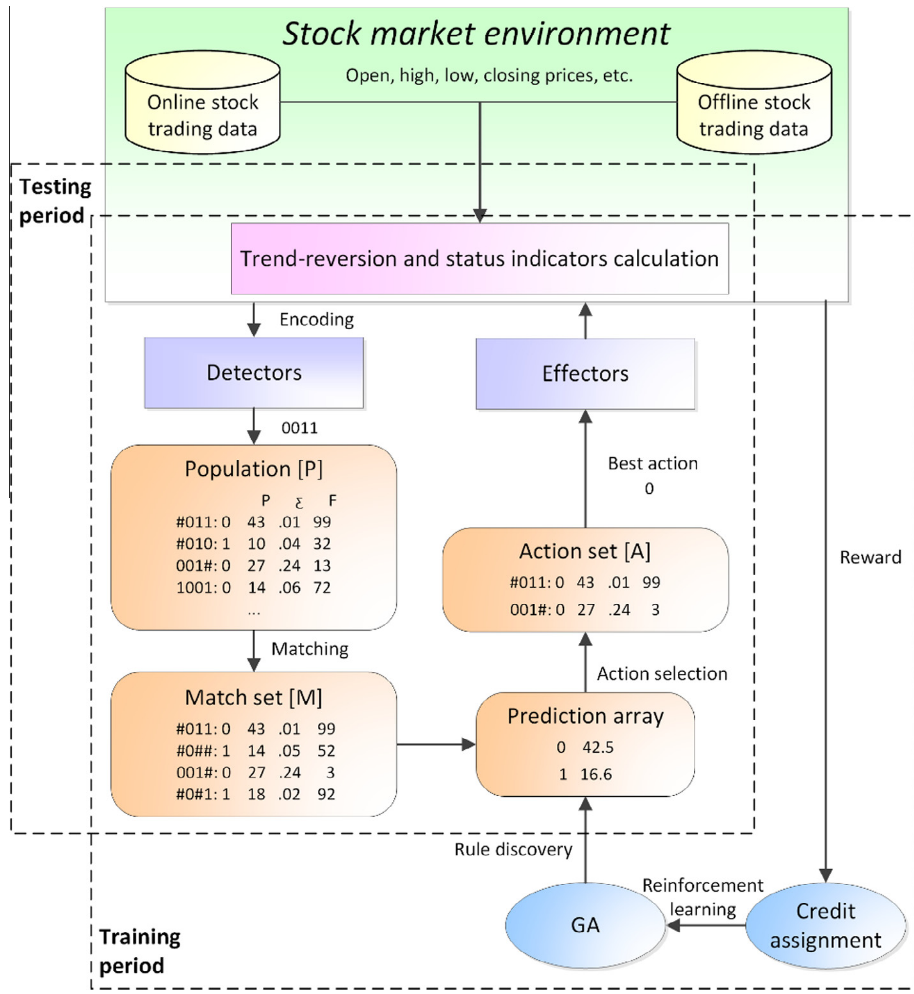

Fig. 1. Schematic illustration of eTrendRev.

to handle the case that no existing classifier matches a given input, as well as a generalization mechanism to reduce the total number of classifiers.

The main difference between training and testing period is that, GA is performed only in the training period. Thus, new classifiers are only generated in training period, while the testing period just utilizes the existing classifiers. Compared to the *pure explore* mode of original XCS, the proposed *learn* mode of XCSL can make adequate use of offline data by exactly learning the optimal action with maximum reward with respect to each historical environment state.

Details of the original XCS could be found in [35] and comprehensive analysis of the system architecture and evolving behavior is in [14].

3.2. XCSL

The original XCS has embedded two action-selection regimes, namely the *pure explore* (random) and *pure exploit* (deterministic) modes [35]. In *pure explore* mode, an action with non-null prediction is selected at random from the prediction array, and both the reinforcement and GA components are invoked. In *pure exploit* mode, the action with the maximum prediction is selected, but the GA component is shut off (except for covering).

Wilson [35] pointed out that “search should be vigorous when little is known or the system is in trouble; once a problem is solved, search is unnecessary” and suggested the dynamic control of the explore/exploit regimes rather than using fixed regimes such as the 50–50 regime used in his paper. In other words, the explore/exploit regime is problem-dependent and search (*pure explore*) is not the optimal strategy when deterministic information is available.

Therefore, this paper introduces the *learn* mode to enhance the original XCS and proposes the XCSL. In stock trading problem, XCSL can make better use of historical price and transaction information in learning how to decide a buy/sell action in future. The pseudo-code of the *learn* mode is provided in Fig. 2. For each action, the historical reward is deterministic and available, and this reward is used to update the classifiers in the corresponding action set. In addition, GA component is triggered to discover new classifiers based on each action set.

3.3. Model process logic

The process logic of eTrendRev is shown in Fig. 3. The logic describes the input and output data flows, the learning and execution processes, etc.

The main difference among the *pure exploit*, *pure explore* and *learn* modes lies on lines 9–20 in Fig. 3. In model testing/trading phase, the *pure exploit* mode selects the action winner that predicts a maximum reward, and does not trigger the GA module (lines 15–19). In model training/learning phase, the *learn* mode repeats lines 9–20 for each action to exhaustively update all internal classifiers (please refer to Fig. 2). However, the *pure explore* mode randomly selects an action as winner and executes lines 9–20 just once.

```
procedure LearnMode
    // Given historical action-reward information for all actions
    for each action a
        Generate action set [A] of a
        Get the reward r of a according to the historical information
        Update classifiers in [A] based on r
        trigger Genetic Algorithm
            Selection
            Crossover
            Mutation
        end trigger
    end for
end procedure
```

Fig. 2. Pseudo-code of the *learn* mode.

```
procedure eTrendRev
    Initialize the population of classifiers [P]
    while (new trading day is available)
        Get stock trading data
        Calculate trend-reversion indicators as environment state
        Decode the state
        Generate match set (Covering if no matched classifier in population) [M]
        Generate prediction array
        Select action winner
        Generate action set [A]
        do winner action
            Get stock return as environment rewards
            Update classifiers in [A]
            trigger Genetic Algorithm
                Selection
                Crossover
                Mutation
            end trigger
        end do
    end while
    Report the learning, execution and profit performances
end procedure
```

Fig. 3. Pseudo-code of eTrendRev.

4. Experiment

In this section, eTrendRev was evaluated in a back-testing experiment. As a reaction to the current stock market states, eTrendRev selects and performs the proper buy/sell action, and then evolves itself according to the subsequent market reward. In addition, eTrendRev is periodically re-trained based on the available historical data, i.e., the previous data before the time of training.

4.1. Datasets

Historical stock price data were used for experiments in this paper. The selected data sets covered a 13-year back-testing period, from January 1, 2001 to December 31, 2013. In total, 3113 and 3239 daily observations were extracted from the SH and NASDAQ indices respectively. Each observation consists of several technical indicators and a return calculated based on closing prices, implying that trading decision is made in the end of each trading day.

4.2. Input variables

The input variables of eTrendRev include the above-mentioned trend-reversion indicators/rules and other status indicators, as shown in Table 1. Parameters of these technical indicators and rules are partly adopted form the previous studies [8,15,20,21].

Only the trend-reversion indicators (i.e., RSI_{buy}, RSI_{sell}, $MACD_{buy}$, $MACD_{sell}$, KD_{buy}, and KD_{sell}) are triggers of buy/sell signals in a theory-driven trend-reversion QI model. Whereas, if eTrendRev only includes these indicators and excludes the other indicators, it would suffer from a non-Markov environment. This limitation of XCS, as previously stated, can be solved by a memory-creating mechanism (e.g., the one used in XCSM and XCSMH) that partly transforms the environment to Markov to choose more informed actions [14]. For better and independent validation of the proposed *learn* mode, the present paper does not adopt XCS variants (e.g., XCSM/XCSMH) for comparison. Instead, the status indicators (i.e., $RSI_{<30}$, $RSI_{>70}$, $MACDS_{>0}$, SD, MA_{asc}, and MA_{des}) are used as a simple trick of memory-creating mechanism.

4.3. Experiment scheme

4.3.1. Sliding window strategy

The sliding window strategy was adopted to divide the sample into training and testing sets, since this method has been widely used in data mining, including stock market forecasting [32]. In this paper, the sizes of training and testing windows were set to be 1000 and 50 (trading days) respectively. In other words, at the beginning of each 50-day trading period, the model was re-trained based on the historical data of previous 1000 trading days before it was used for trading decision.

Moreover, input variables were transformed into a 0/1 bit strings used as environment states in the XCSL, while next-day returns were used as rewards of actions fed back by environment.

4.3.2. Transaction costs

In practical trading, transaction cost must be taken into consideration, since it accounts for an important part of expense in technical analysis trading, especially in high-frequency trading [1,27]. The following experiment calculates transaction costs based on the unique commission fee and transaction tax. Specifically, it is 0.1% of the transaction capital for buy (long position) and 0.2% for sell (short position) [1,2].

4.3.3. Performance measures

In terms of prediction accuracy, hit ratio was adopted to evaluate the success rate of up/down-trend prediction, i.e., positive/negative return prediction.

In terms of investment profit, common evaluation indicators including accumulated return ratio, annualized return ratio, excess return ratio, as well as the buy-and-hold return ratio were calculated.

In terms of investment risk, Sharpe ratio [29], Sortino ratio [30] and maximum drawdown ratio were presented. Sharpe ratio measures the average excess return on a risk-free return per unit of standard deviation of a trading strategy. A higher return and lower volatility leads to higher Sharpe ratio. Sortino ratio differs with Sharpe ratio in the calculation of risk. Sortino ratio associates risk only to returns that are below the minimal acceptable value/level, and thus it estimates the average excess return relative to the downside deviation (by using the lower partial moment approach). Finally, the maximum drawdown ratio indicates the maximum asset value drop from a peak to bottom during the whole period, which measures how sustained one's losses can be.

4.4. Result

4.4.1. Comparison between the Learn and Pure explore modes

To examine whether the proposed *learn* mode is superior to the original *pure explore* mode, the trading returns of eTrendRev with either XCSL or XCS in simulation experiments were compared. Because the GA operations would generate different results in each simulation, back-testing was carried out 20 times. After eliminating the highest result as well as the lowest one, the remaining 18 results were used for statistical analysis. Table 2 lists the comparative result between these two modes over the SH and NASDAQ Indices.

This result implies that the *learn* mode is more stable than the *pure explore* mode. On the SH Index, the *learn* mode obtained average return of 205.65%, which is higher than that of the *pure explore* mode (183.95%). Besides, returns of the *learn* mode covered a range from 158.07% to 321.83% with a standard deviation of 49.04%, while the range of returns covered by the *pure explore* mode was from 101.81% to 278.32% with a standard deviation of 55.21%. This indicates that the *learn* mode prevails in all aspects. On the NASDAQ Index, the *learn* mode still performed better in most aspects, except the *pure explore* mode obtained a higher maximum return of 77.85% than that of the *learn* mode (66.76%). This may be due to the random-exploring ability of the *pure explore* mode that has extended the search space and "occasionally" uncovered some outstanding rules.

In conclusion, the comprehensive performance analysis shows that the proposed *learn* mode is superior to the original *pure explore* mode.

4.4.2. Risk and return evaluation

In terms of risk and return, detailed performance of the best model, which is generated by eTrendRev with XCSL on the SH Index, are presented in Table 3 and Fig. 4. We also compared the performance to two classical data mining algorithms that can generate explicit rule knowledge – C4.5 decision tree [27] and Bayesian network (BN) [38].

The return rate of eTrendRev was 321.83% (or an annualized return rate of 18.57%), compared to 76.06% from a buy-and-hold strategy. And the risk was acceptable since the Sharpe ratio and Sortino ratio reached 0.82 and 1.18 respectively, and the maximum drawdown ratio was 23.27%. With the same input variables and back-testing settings, C4.5 generated much worse result, with lower hit ratio, return ratio, Sharpe ratio and Sortino ratio, and higher maximum drawdown ratio, which implies that C4.5 suffers from lower return and higher risk. In addition, BN generated the worst result and underperformed the buy-and-hold strategy.

Short-selling was not considered in this experiment setting because it is not allowed in the Chinese stock market. Thus, one can only make profit by buying an up-trend stock, while selling a down-trend stock can only avoid loss. As shown in Fig. 4, the SH Index declined at most of the time, and eTrendRev released successive sell signals in these long-term down-trends, which made no profit. If short-selling was allowed, the performance would be much better.

4.4.3. Trading signal analysis

To evaluate whether eTrendRev can identify significant market turning points, in other words, whether the selected

Table 1
Input variables of eTrendRev.

Variable	Definition
RSI_{buy}	$RSI(6)_{t-1} \leqslant 30$ and $RSI(6)_t > 30$
RSI_{sell}	$RSI(6)_{t-1} \geqslant 70$ and $RSI(6)_t < 70$
$RSI_{<30}$	$RSI(6)_t < 30$
$RSI_{>70}$	$RSI(6)_t > 70$
$MACD_{buy}$	$MACD(12,26)_{t-1} \leqslant MACDS(12,26,9)_{t-1}$ and $MACD(12,26)_t > MACDS(12,26,9)_t$
$MACD_{sell}$	$MACD(12,26)_{t-1} \geqslant MACDS(12,26,9)_{t-1}$ and $MACD(12,26)_t < MACDS(12,26,9)_t$
$MACDS_{>0}$	$MACDS(12,26,9) > 0$
KD_{buy}	$\%K(9)_{t-1} \leqslant \%D(3)_{t-1}$ and $\%K(9)_t > \%D(3)_t$ and $\%K(9)_t > 20$
KD_{sell}	$\%K(9)_{t-1} \geqslant \%D(3)_{t-1}$ and $\%K(9)_t < \%D(3)_t$ and $\%K(9)_t < 80$
SD	$\sigma([MA_5, MA_{15}, MA_{30}])_t > \bar{\sigma}([MA_5, MA_{15}, MA_{30}])_{t-1}$
MA_{asc}	$MA(5) > MA(15) > MA(30)$
MA_{des}	$MA(5) < MA(15) < MA(30)$

Table 2
Statistics of trading returns in simulation experiments.

Dataset	Mode	Min (%)	Max (%)	Avg. (%)	Sd. (%)
SH	*Pure explore*	101.81	278.32	183.95	55.21
	Learn	**158.07**	**321.83**	**205.65**	**49.04**
NASDAQ	*Pure explore*	14.66	**77.85**	40.08	15.13
	Learn	**20.67**	66.76	**44.18**	**13.99**

Note: In each dataset, the highest and the lowest results were eliminate and only 18 results were reserved for statistics.
The better result between *Pure explore* and *Learn* modes is in bold.

Table 3
Detailed performance comparison on SH Index.

Performance	eTrendRev	C4.5	BN
Hit ratio	57.38%	55.01%	51.34%
Return rate	321.83%	148.54%	41.67%
Annualized return rate	18.57%	11.37%	4.21%
Sharpe ratio	0.82	0.49	0.16
Sortino ratio	1.18	0.70	0.22
Maximum drawdown ratio	23.27%	44.01%	54.87%
Buy-and-hold return ratio	76.06%	76.06%	76.06%

trend-reversion indicators are effective, it is necessary to investigate (1) when and where a turning point occurs, (2) whether this point is captured by eTrendRev with proper buy/sell signal, and (3) how well the signal is generated (i.e., with delay or in advance). Several trading signals on the SH Index (in accordance with the result presented in Table 3) are taken for demonstration, as shown in Fig. 5.

In Fig. 5, several significant turning-points are indicated by dashed lines, where buy signals are indicated by red "B" and sell signals by green "S". Detailed analysis is presented in chronological order (from left to right on the *x*-axis) as follows:

(1) The first dashed line is on Dec 20, 2007, and successive sell signals were generated. The SH Index had already dropped a significant amount (about 16%) from the highest peak of the whole period (on Oct 16, 2007). Although these turning-point signal delayed for about 2 months, eTrendRev had avoided the subsequent long-term decline (about 72%) that lasted for about 1 year, and only a few of sparse buy signals were generated in this period.
(2) The second dashed line is on Sep 19, 2008, where a series of scattered buy signals were generated. Although these signals were released about 2 months before the significant turning-

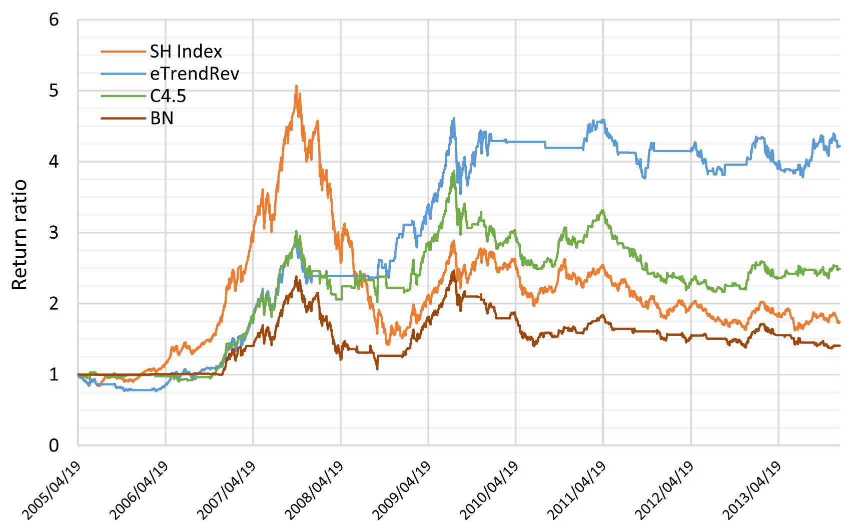

Fig. 4. Yield curve comparison on SH Index.

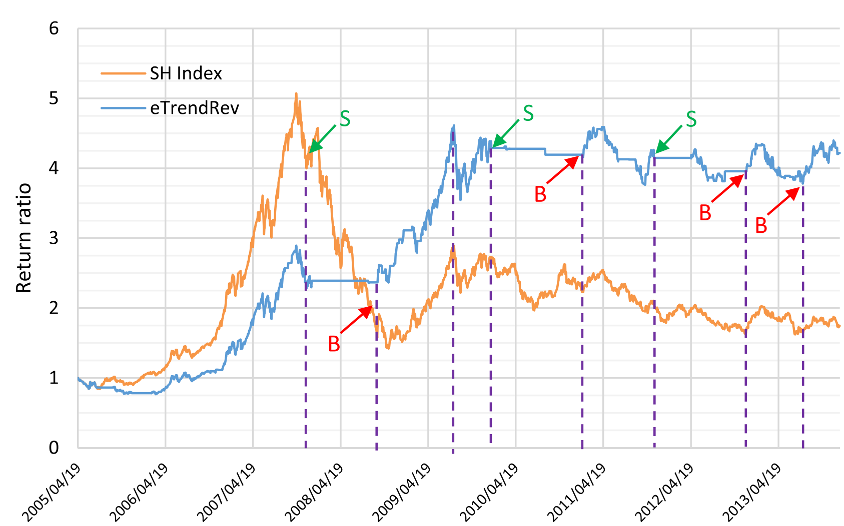

Fig. 5. Several significant trading signals on the SH Index.

point on Nov 6, 2008 (an important bottom), the yield curve was rising at most of the time in this period. After this market bottom, sparse buy signals made significant profit (about 21%), followed by successive buy signals from Mar 18, 2009 to Jan 4, 2010 which resulted in a profit rise of 45.7%. From Sep 19, 2008 to July 28, 2009 (a peak on the third dashed line), the buy-and-hold return on the SH Index was 53.8% while eTrendRev earned 79.3%. However, eTrendRev could not predict the fast and drastic price reversal ranged from Aug 5, 2009 to Aug 29, 2010 and suffered a total loss about 22.2%.

(3) The fourth dashed line is on Jan 7, 2010, and successive sell signals (only with a few buy signals) were generated until Jan 24, 2011 (the fifth dashed line). During this period, a great loss about 47.8% was avoided.

(4) The fifth dashed line is on Jan 24, 2011, and a series of buy signals were generated to capture a 3-month up-trend just before it started. However, a large part of the following decline could not be avoided, though eTrendRev still outperformed the SH Index during this period from Jan 24, 2011 to Nov 16, 2011 (the six dashed line). Then successive sell signals were released to avoid further decline.

(5) The seventh dashed line is on Dec 6, 2012. A series of buy signals were generated to capture the subsequent 6-month up-trend, just with a 3-day delay after the market bottom on Nov 30, 2012. Similar situation recurred around the eighth and last dashed line on July 16, 2013.

5. Conclusion

To cope with the challenge of big data analysis in QI, this paper proposes a model called eTrendRev, which integrates the proposed XCSL with trend-reversion strategy. In the context of 4V in big data era, eTrendRev can adapt itself to the varying stock market environment owing to the evolutionary and reinforcement learning capabilities. Experimental results showed that eTrendRev can generate effective buy/sell signals and outperform the buy-and-hold strategy with high Sortino ratio after transaction cost on the selected SH Index.

Further findings include: (1) eTrendRev is less sensitive to short-term violent fluctuation than long-term trends. However, by adjusting the period parameters of input variables, it might be able to balance between stability and flexibility, namely, between capturing long-term trends and reacting to short-term fluctuations. (2) The trend-following ability of eTrendRev needs further improvement, even though most long-term up-trends are confirmed and followed. This result is not surprising because only trend-reversion indicators were adopted in eTrendRev.

Contributions of this study include: (1) proposing the *learn* mode that performs better and is more stable than the original *pure explore* mode of XCS, and (2) confirming the profitability of using only the selected trend-reversion indicators in ML-based QI model.

Limitation of this study mainly lies on the fact that the RSI, stochastic (K&D), and MACD adopted in this paper are just a smattering of the numerous trend-reversion indicators available in literature and practical trading, such as BAIS, psychological line [20,21]. From the practical standpoint, it is difficult to incorporate all existing indicators and test every combination of indicator parameters. Moreover, other sources of big data, such as abundant news and/or comments released in the Internet and various social media, can be considered to enhance the environment perception ability of eTrendRev.

Future study can focus on incorporating more intuitional measures for each classifier in XCS, since the original prediction, prediction error and fitness measures are defined in an inscrutable way. For example, the *support* and *confidence* measures from association analysis [7,34] are suitable and can be adapted to XCSL by considering condition-state matching and the level of reward (positive/negative). Such an improvement is similar to the success rate-based experience-evaluation mechanism of SRXCS [28]. In addition, most of existing XCS-based stock models applied the original XCS without significant improvement, though, in other domains, a variety of well-designed XCS variants and extensions have been proposed, such as those in [28,31,37]. Therefore, as future study, we can apply these advanced XCS techniques.

Acknowledgments

This research was partly supported by the National Natural Science Foundation of China (71271061, 70801020), Major Project of National Social Science Foundation of China (14ZDA074), Science and Technology Planning Project of Guangdong Province, China (2010B010600034, 2012B091100192), and The Ministry of Education Innovation Team Development Plan, Guangdong Natural Science Foundation Research Team, and Business Intelligence Key Team of Guangdong University of Foreign Studies (S2013030015737, IRT1224, TD1202).

References

[1] F. Allen, R. Karjalainen, Using genetic algorithms to find technical trading rules, J. Financ. Econ. 51 (1999) 245–271.
[2] J. Andrada-Félix, F. Fernández-Rodríguez, Improving moving average trading rules with boosting and statistical learning methods, J. Forecast. 27 (2008) 433–449.
[3] R.J. Balvers, Y. Wu, Momentum and mean reversion across national equity markets, J. Empir. Financ. 13 (2006) 24–48.
[4] Z. Bodie, A. Kane, A. Marcus, Investments, sixth ed., McGraw-Hill, New York, 2005.
[5] M.A. Boyacioglu, D. Avci, An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: the case of the Istanbul stock exchange, Expert Syst. Appl. 37 (2010) 7908–7912.
[6] Q. Cao, K.B. Leggio, M.J. Schniederjans, A comparison between Fama and French's model and artificial neural networks in predicting the Chinese stock market, Comput. Oper. Res. 32 (2005) 2499–2512.
[7] Y. Chang Chien, Y. Chen, Mining associative classification rules with stock trading data – a GA-based method, Knowl.-Based Syst. 23 (2010) 605–614.
[8] P. Chang, C. Fan, A hybrid system integrating a wavelet and TSK fuzzy rules for stock price forecasting, IEEE Trans. Syst. Man Cyber. C – Appl. Rev. 38 (2008) 802–815.
[9] A. Chen, M. Chen, Integrating extended classifier system and knowledge extraction model for financial investment prediction: an empirical study, Expert Syst. Appl. 31 (2006) 174–183.
[10] M. Covel, Trend Following: How Great Traders Make Millions in Up or Down Markets, first ed., Financial Times Prentice Hall, New Jersey, 2004.
[11] J. Cunado, L.A. Gil-Alana, F.P.D. Gracia, Mean reversion in stock market prices: New evidence based on bull and bear markets, Res. Int. Bus. Financ. 24 (2010) 113–122.
[12] S. Fong, J. Tai, Y.W. Si, Trend following algorithms for technical trading in stock market, J. Emerg. Technol. Web Intel. 2 (2011) 136–145.
[13] S. Fong, Y. Si, J. Tai, Trend following algorithms in automated derivatives market trading, Expert Syst. Appl. 39 (2012) 11378–11390.
[14] D.A. Gaines, R. Pakath, An examination of evolved behavior in two reinforcement learning systems, Decis. Support Syst. 55 (2013) 194–205.
[15] A. Gorgulho, R. Neves, N. Horta, Applying a GA kernel on optimizing technical analysis rules for stock picking and portfolio composition, Expert Syst. Appl. 38 (2011) 14072–14085.
[16] R.G. Hagstrom, The Warren Buffett Way: Investment Strategies of the World's Greatest Investor, first ed., John Wiley & Sons, New York, 1997.
[17] J.H. Holland, Adaptation, in: F.M. Snell, R. Rosen (Eds.), Progress in Theoretical Biology, Academic Press, New York, 1976, pp. 263–293.
[18] Y. Hsu, A. Chen, J. Chang, An inter-market arbitrage trading system based on extended classifier systems, Expert Syst. Appl. 38 (2011) 3784–3792.
[19] M. Kaucic, Investment using evolutionary learning methods and technical rules, Eur. J. Oper. Res. 207 (2010) 1717–1727.
[20] L.N. Kestner, Quantitative Trading Strategies: Harnessing the Power of Quantitative Techniques to Create a Winning Trading Program, first ed., McGraw-Hill, New York, 2003.

[21] R.K. Lai, C. Fan, W. Huang, P. Chang, Evolving and clustering fuzzy decision tree for financial time series data forecasting, Expert Syst. Appl. 36 (2009) 3761–3773.
[22] M. Lam, Neural network techniques for financial performance prediction: integrating fundamental and technical analysis, Decis. Support Syst. 37 (2004) 567–581.
[23] P.L. Lanzi, An Analysis of the Memory Mechanism of XCSM, Morgan Kaufmann, San Francisco, California, 1998. pp. 643–651.
[24] P.L. Lanzi, S.W. Wilson, Toward optimal classifier system performance in non-Markov environments, Evolut. Comput. 8 (2000) 393–418.
[25] S. Mitra, T. Acharya, Data Mining: Multimedia, Soft Computing, and Bioinformatics, first ed., John Wiley & Sons, New York, 2005.
[26] R.K. Narang, Inside the Black Box: The Simple Truth about Quantitative Trading, second ed., John Wiley & Sons, Hoboken, New Jersey, 2013.
[27] N. Ren, M. Zargham, S. Rahimi, A decision tree-based classification approach to rule extraction for security analysis, Int. J. Inform. Technol. Decis. Making 5 (2006) 227–240.
[28] M. Shariat Panahi, A. Karkhaneh Yousefi, M. Khorshidi, Combining accuracy and success-rate to improve the performance of eXtended Classifier System (XCS) for data-mining and control applications, Eng. Appl. Artif. Intel. 26 (2013) 1930–1935.
[29] W.F. Sharpe, The Sharpe ratio, J. Portfolio Manage. 21 (1994) 49–58.
[30] F.A. Sortino, L.N. Price, Performance measurement in a downside risk framework, J. Invest. 3 (1994) 59–64.
[31] M. Troć, O. Unold, Self-adaptation of learning rate in XCS working in noisy and dynamic environments, Comput. Hum. Behav. 27 (2011) 1535–1544.
[32] C. Tsai, Y. Hsiao, Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches, Decis. Support Syst. 50 (2010) 258–269.
[33] W. Tsai, A. Chen, Using the XCS classifier system for portfolio allocation of MSCI index component stocks, Expert Syst. Appl. 38 (2011) 151–154.
[34] J. Wei, W. Yi, M. Wang, Novel measurement for mining effective association rules, Knowl.-Based Syst. 19 (2006) 739–743.
[35] S.W. Wilson, Classifier fitness based on accuracy, Evolut. Comput. 3 (1995) 149–175.
[36] L. Yu, H. Chen, S. Wang, K.K. Lai, Evolving least squares support vector machines for stock market trend mining, IEEE Trans. Evolut. Comput. 13 (2009) 87–102.
[37] Z. Zang, D. Li, J. Wang, D. Xia, Learning classifier system with average reward reinforcement learning, Knowl.-Based Syst. 40 (2013) 58–71.
[38] Y. Zuo, E. Kita, Stock price forecast using Bayesian network, Expert Syst. Appl. 39 (2012) 6729–6737.