CSCanada
Canadian Social Science
Vol. 12, No. 11, 2016, pp. 89-95
DOI:10.3968/9006

ISSN 1712-8056[Print]
ISSN 1923-6697[Online]
www.cscanada.net
www.cscanada.org

Convexity-Concavity Indicators and Automated Trading Strategies Based on Gradient Boosted Classification Trees Models

ZHANG Qun[a],*

[a]School of Business Administration, South China University of Technology, Guangzhou, China.
*Corresponding author.

Received 5 August 2016; accepted 8 October 2016
Published online 26 November 2016

Abstract

This paper uses the visibility and invisibility algorithms to build the peak and trough indicators, providing a way to recognize the convexity, concavity and regime change of the CSI 300 Index from the April 8, 2005 to June 30, 2016. The study found that the automated trading rules discovered by the gradient boosted classification trees models derived from the peak indicator outperform that from the trough indicator. Due to the long-term bubble regime in the Chinese stock market, the technical trading rules in general have a better short term predictive ability than long term, in terms of the values of Sharpe Ratio and PnL/MD obtained from the whole out-of-sample.

Key words: Trading strategy; Machine learning; Boosting; Binary classication; Regime shift

Zhang, Q. (2016). Convexity-Concavity Indicators and Automated Trading Strategies Based on Gradient Boosted Classification Trees Models. *Canadian Social Science, 12*(11), 89-95. Available from: http://www.cscanada.net/index.php/css/article/view/9006 DOI: http://dx.doi.org/10.3968/9006

INTRODUCTION

The stock market is considered as a high complex and dynamic system with noisy, non-stationary and chaotic time series. The intricate issue of nowcasting and forecasting the phase transition has been one more time well-illustrated recently with the global downturn experienced by most countries around the world. Governments, central banks and investors are very sensitive to financial indicators showing readable signals in order to adjust their policies or technical strategies sufficiently to gain profit or avoid loss in advance (Park & Irwin, 2007).

The method of technical analysis aims to predict stock returns by finding persistent patterns and relationships generalized beyond the historical data (Allen & Karjalainen, 1999; Friesen et al., 2009). Except for incorporating historical movements to discover technical trading rule, another high-level research is detecting the trend of phase transition in financial markets (Wen et al., 2010). Various soft computing techniques have been introduced for early detection or prediction purpose among both academics and industry professionals over the last two decades. A particularly active area of research is employing artificial intelligence methods (Booth et al., 2014) or evolutionary computation techniques (Hu et al., 2015) to recognise investment opportunities and reduce error derived from noise, with which is better than or at least as good as their human counterparts. Trading rule discovery technique could be a way to integrate the information derived from relevant technical indicators, filter the noise and increase the return adjusted to the risk (Cervello-Royo et al., 2015).

Convexity is an important concept that gains insight into the more technical aspects accumulated in stock investment (Kwon & Moon, 2007). Understanding this basic characteristic allows the investor to better comprehend the evolution of financial time series. Empirical observations associated with the Log-periodic Power Law Singularity model have revealed the characterisation of super exponential growth in price time series before the crashes (Johansen & Sornette, 1999; Johansen et al., 2000; Sornette, 2009). Forecasting financial extremes thus refers to the detection and measure of the occurrence of this kind of growth. However, it is an experience-dependent to make trading decisions for the difficulties in identifying the robust signals and measuring the effectiveness.

Gradient boosting is widely used by data scientists to achieve state-of-the-art results on many machine learning challenges (Chen & Guestrin, 2016). Differ from linear models like logistic regression or support vector machine, gradient boosted trees can model non-linear interactions between the features and the target. The gradient boosted trees model has become one of the most effective machine learning models. It is not only suitable for handling numerical features and categorical features with tens of categories, but also can be used as a classifier for predictive tasks. The prediction is based on a collection of base learners (i.e., decision tree classifiers) and combines them through a technique called gradient boosting.

In this paper, it aims to exploit a convexity-concavity representation of time series and use the gradient boosted classification trees models to discover trading rules. The basic idea is that the stock price time series is supposed to exhibit certain structures inherited in a period of time. It thus employs the visibility and invisibility algorithm to convert a time series into a price network (Lacasa et al., 2008), and builds the peak and trough indicators to measure the magnitude of the super-exponential growth of stock prices (Yan & Serooskerken, 2015). The strengths of the approach lie in two main aspects. First, the convexity-concavity indicators specifically measure all conceivable upward or downward growth laws that are faster than exponential. The actual values of the indicators thus may foretell the future. Second, trading signals that suggest buy or sell are discovered by the ensemble learners of the gradient boosted classification trees models for a higher prediction performance.

1. PEAK AND TROUGH INDICATORS

Yan and Serooskerken (2015) constructed peak and trough indicators as two financial extreme indicators to predict the peaks and troughs in the financial time series. For a log-price time series, $y_i = \log(p_i)$, the peak degree $D_{\text{peak}}(t_i)$ at time t_i measures the number of data points (t_i, y_i) in the time window $t_j \in [t_{i-s}, t_{i-1}]$ satisfying:

$$y_i > y_j, \tag{1}$$

$$y_k < y_j + \frac{t_k - t_j}{t_i - t_j}(y_i - y_j), \forall j < k < i, if\ \ i - j \geq 2. \tag{2}$$

The peak indicator at time t_i is defined as $PI(t_i) = \frac{D_{\text{peak}}(t_i)}{S} \in [0,1]$.

Similarly, the trough indicator at time t_i is defined as $TI(t_i) = -\frac{D_{\text{trough}}(t_i)}{S} \in [-1,0]$. The absolute value $|TI(t_i)|$ measures the proportion of data points (t_i, y_i) in the time window $t_j \in [t_{i-s}, t_{i-1}]$ satisfying:

$$y_i < y_j, \tag{3}$$

$$y_k > y_j + \frac{t_k - t_j}{t_i - t_j}(y_i - y_j), \forall j < k < i, if\ \ i - j \geq 2. \tag{4}$$

Thus, the peak indicator *PI* is a measure of convexity of the log-price as a function of time at a local maximum. Hence it is a direct measure of the super-exponential acceleration. At the same time, it can be influenced by the volatility of the log-price decorating the convexity. So a large peak indicator indicates: there is a local maximum such that the log-price has shown a strong convexity over *S* trading days while the volatility during this upward acceleration has been moderate.

Similarly, the trough indictor *TI* is a measure of concavity of the log-price as a function of time at a local minimum. It also can be influenced by the roughness of the log-price decorating the concavity.

2. APPLICATIONS TO THE REAL DATA

This section thus applies above measures to investigate the convexity and concavity of the CSI 300 Index.

2.1 Data

The CSI 300 is a capitalization-weighted stock market index designed to replicate the performance of 300 stocks traded in the Shanghai and Shenzhen stock exchanges. The data to be examined from the April 8, 2005 to June 30, 2016 are from the Thomson Reuters Datastream.

2.2 Calculation of Two Indicators

Three panels in Figure 1 illustrate how to respectively calculate two indicators of CSI 300 by the visibility algorithm and absolute invisibility algorithm at each t_i from the January 3, 2012 to June 30, 2016.

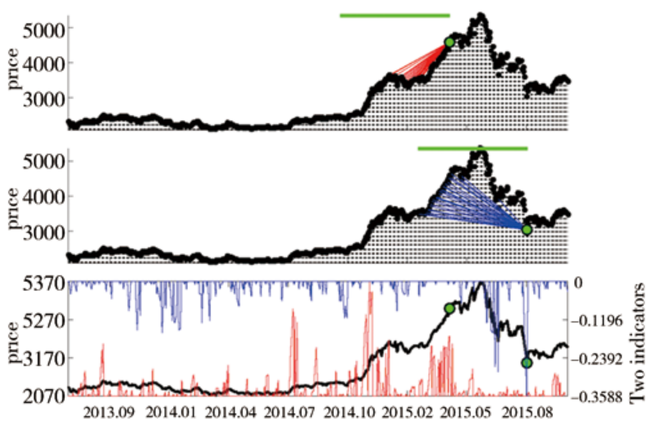


Figure 1
Calculation of Peak and Trough Indicators of CSI 300

As the red lines shown in the top panel of the Figure 1, for the peak indicator *PI* at t_i=2015.04.17, 31 data points are linked to the green point in the look-back window with a length of *S*=131 trading days (represented by the green line) that satisfying the Equation (1) and Equation (2) by the visibility algorithm. Thus, $PI(t_i$=2015.04.17) $= \frac{31}{131}$ =0.237. For the trough indicator *TI* at t_i=2015.08.26 in the second panel, 29 data points are linked to the green point in the look-back window with a length of S=131 trading days (represented by green line) that satisfying the Equation (3) and Equation (4) by the absolute invisibility algorithm. Thus, $TI(t_i=2015.08.16) = -\frac{29}{131} = -0.211$. The bottom panel of Figure 1 further shows the performance of peak indicator (red line) and trough indicator (blue line) in the window [2012.01.03, 2016.06.30].

2.3 Performance of Two Indicators on Real Data

Figure 2 presents the performance of the two indicators of CSI 300 in the whole window [2005.04.08, 2016.06.30]. Four panels show the price time series (black line in the top panel), the signals of *PI* (red line in the second panel), the signals of *TI* (blue line in the third panel), and the cross-correlation of these two discrete signal sequences as a function of the lag ranged from -131 to 131 (black dot in the bottom panel).

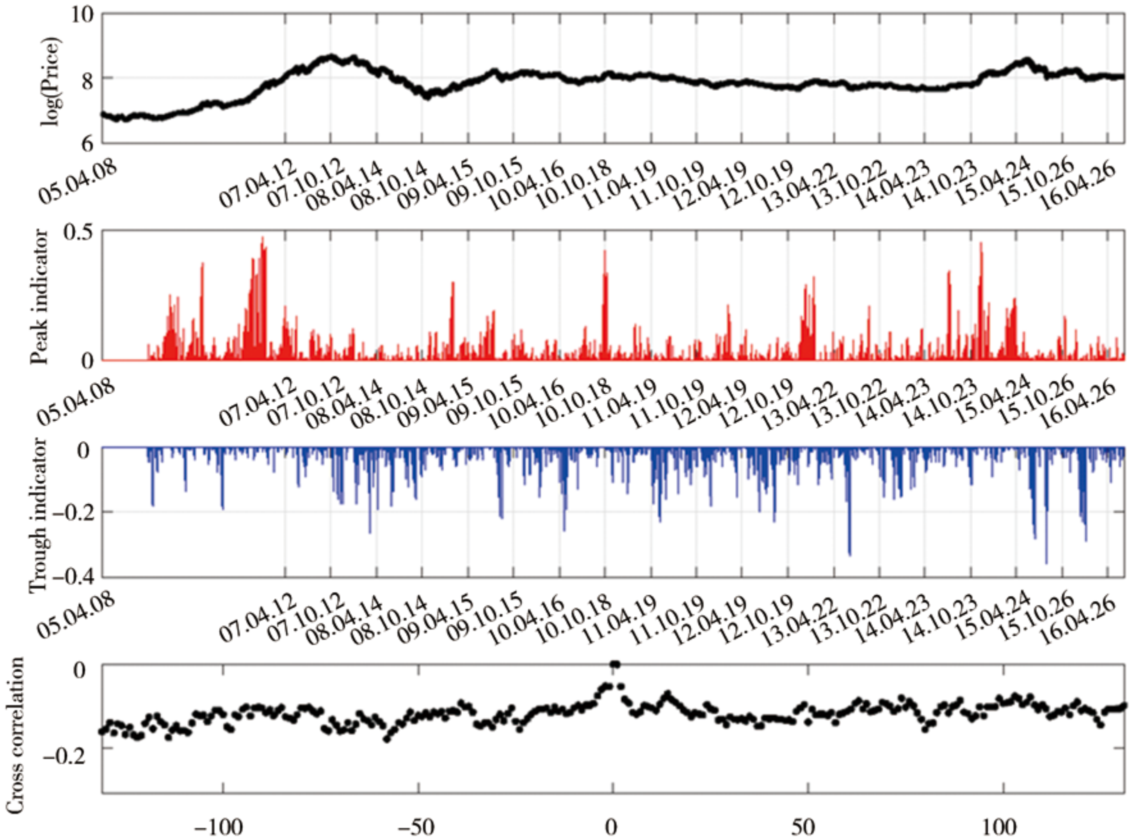


Figure 2
CSI 300 Time Series and Signals of *PI* and *TI* and Their Cross-Correlation

The results shown in Figure 2 are obtained using the window size S=131 trading days, i.e., roughly half a calendar year. The peaks of signals in the second and third panels in the Figure 2 show that most of the change of regime can be diagnosed by *PI* and *TI*. However, the reality is so complex that the indicators have been influenced by the volatility of time series. The different height of bunches of signals continually derived from *PI* means that the log-price convex ups at different rates. Some of them are even followed by or mixed with the concave-up sharp until reaching the critical time. And their cross-correlation values within the range of lag from -131 to 131 are very low. They thus are suitable to be two features of the gradient boosted classification trees model for capturing the states of market.

3. AUTOMATED TRADING STRATEGIES BASED ON GRADIENT BOOSTED CLASSIFICATION TREES MODELS

Given the historical data, this section thus trains the data to develop profitable strategies based on the gradient boosted classification trees models, and tests the aforementioned indicators' performance and usability.

3.1 Defining the Peaks and Troughs

First, it searches the local "peak" (noted as PK) as defined in Equation (5) and local "trough" (noted as TR) in Equation (6) along the whole time series, where b and a determine the length of searching windows (Yan & Serooskerken, 2015).

$$PK=(t_i, y_i),\text{where } y_i = \max(y_j), \quad (5)$$

$$\text{TR}=(t_i, y_i), \text{ where } y_i=\min(y_j), \quad i-b \le j \le i+a, j \in \mathbf{N}. \tag{6}$$

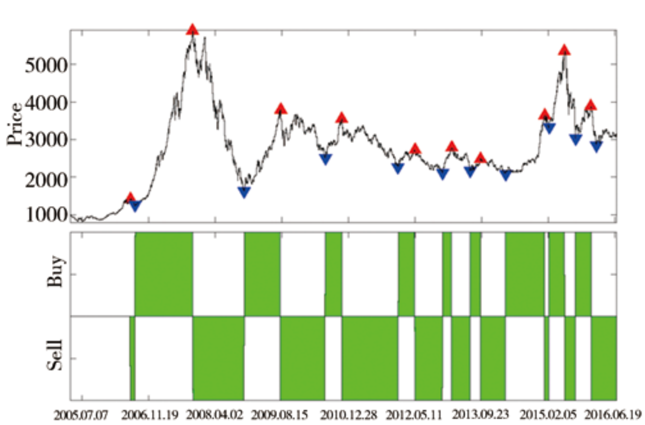
3.2 Trading in the Well-Defined Regimes

The found local peaks and troughs are the points dividing the whole time series as continuous regimes. For an interval [PK, TR] therein with a peak as its left point and a trough on the right, one of the simple trading strategies is selling the asset at the beginning and buying it at the end. The profit in the interval [TR, PK] can also be realized from buying the asset at first and selling it later. For the interval [PK1, PK2], it further finds the minimum in it as a local trough that divides the whole into two subintervals, such that sell in the interval [PK1, TR] and shift to hold it in the interval [TR, PK2]. For an interval [TR1, TR2], a new found local peak PK would divide the regime to [TR1, PK] for holding and [PK, TR2] for selling.

Given that b=131 and a=45 trading days, the top panel of Table 1 presents all the found peaks (red triangle) and troughs (blue inverted triangle) of CSI 300 and the corresponding benchmark trading strategy. The statistical measures at the bottom give a reference being the best that can be achieved, without a consideration of transaction costs. So these buy-sell signals can be the input of target to train trees models. The statistical measures can also be used to examine the performance of out-of-sample. For the Maximum Drawdown (MD), a lower output means a better performance. With the Sharpe Ratio, Annual Return (PnL) or PnL/MD, a higher output is better.

Table 1
Benchmark Trading Strategy and Its Performance of CSI 300

Found peaks and troughs and benchmark trading strategy



Statistical measures	Values
Sharpe Ratio	2.03
Annual Return (PnL)	0.41
Maximum Drawdown (MD)	0.17
PnL/MD	2.41
Trading days	2406.00
Trade counts	9.00

3.3 Representation of Gradient Boosted Classification Trees Model

3.3.1 Input

Feature vector: $X=\{x_{p,i}, p=1,2; i=1,2,...,N\}=\{PI(t_i), TI(t_i), i=1,2,...,N\}$. Please note that the absolute value of $TI(t_i)$ is used during the training for a better comparison with $PI(t_i)$.

Target for training: $Z=\{z_i, i=1,2,...,N\} \in \{-1,1\}$, that is, the buy-sell signals derived from the benchmark trading strategy in Table 1.

Goal:

Given the training data (X,Z), produce a binary classifier $F(X)=\text{sign}[f_M(X)]=\text{sign}\left[\sum_{m=1}^{M} v\beta_m b(X,\gamma_m)\right] \in \{-1,1\}$ in a form of a weighted sum of tree $b(X,\gamma_m)$ where γ_m parametrizes the splits.

3.3.2 Algorithm

(a) Initialize the model with a constant value: $f_0(X)=0$;

(b) From m=1 to M repeat: compute $(\beta_m,\gamma_m)=\arg\min_{\beta,\gamma}\sum_{i=1}^{N} L(z_i, f_{m-1}(x_{p,i})+\beta b(x_{p,i},\gamma))$, where the loss function $L(\bullet)$ not only measures how well model fit the training data, but also measures the complexity of model mainly through the term $\beta b(x_{p,i}, \gamma)$;

(c) Update the model: $f_m(X)=f_{m-1}(X)+v\beta_m b(x_{p,i}, \gamma)$.

3.3.3 Key Parameters

- M(*max iterations*): The maximum number of iterations for training. Each iteration results in the creation of an extra tree.
- v(*step size*): Step size (shrinkage) used for combining the weight of individual trees in update to prevents overfitting. It shrinks the prediction of each weak learner to make the boosting process more conservative. The smaller the step size, the more conservative the algorithm will be. Smaller v works well when M is large, that is, the lower "learning rate" requires more iteration.
- β_m(*min loss reduction*): Minimum loss reduction required to make a further partition on a leaf node of the tree. The larger the non-negative value is, the more conservative the algorithm will be.
- γ_m(*max depth; min child weight*):

max depth: The maximum depth of the individual decision trees (at least 1). It not only controls the number of terminal nodes in trees, but also controls the maximum allowed level of interaction between variables in the model. If it equals to 2, there is no interaction between variables allowed. When it equals to 3, the model may include effects of the interaction between up to two variables, and so on.

min child weight: This non-negative parameter controls the minimum weight assigned to leaf nodes. The larger it is, the more conservative the algorithm will be. Formally, this is minimum sum of instance weight (hessian) in each leaf. If the tree partition step results in a leaf node with the sum of instance weight less than it, then the building process will give up further partitioning.

In the following tests, the parameters are set as
max iterations=50, step size=0.01,
min loss reduction=1, max depth=3,
min child weight=0.6.

3.3.4 Evaluation and Validation

The correctness of a classification can be evaluated by computing the number of correctly recognized class examples (true positives), the number of correctly recognized examples that do not belong to the class (true negatives), and examples that either were incorrectly assigned to the class (false positives) or that were not recognized as class examples (false negatives).

The metric accuracy (Sokolova & Lapalme, 2009) can be used to measure the fraction of predictions of target values made by the classifier that are exactly correct. The score lies in the range [0, 1] with 0 being the worst and 1 being the best. In the following tests, the classification accuracy is measured both on the training data and validation set through:

$$\text{accuracy}=\frac{\text{true positives} + \text{true negatives}}{\text{true positives} + \text{true negatives} + \text{false positives} + \text{false negatives}}. \quad (7)$$

3.3.5 Output

Classification trees use a tree structure to recursively partition the features space until the subsets of feature space are tame enough to fit simple models to them. Each of the leaf nodes of a tree represents a small subset of the feature space, and is attached to a simple model which applies only to that subset. The cell to which a data point belongs is identified by starting at the root node of the tree, and answering a sequence of questions about the feature values. Measuring these sets of possible outcomes thus provides a way to select the features and develop effective trading strategies.

A series of binary spits that each internal node represents a value query on one of the variables, while the terminal nodes are the decision nodes, and classified into two classes taking values +1 (buy signal) or -1 (sell signal) with the estimated aggregate score $f_M(X)$.

3.4 Classification-Based Trading Strategy

Figure 3 shows the classification process towards a higher PnL/MD in three steps.

In step 1, it finds the local peaks and troughs along the whole time series determined by *b* and *a*. For each round of training and testing in step 2, data are divided into three parts, with the length of InLen, *a* and OutLen. Since *a* trading days' data are needed to define the peaks and troughs that further determine the input of target *Z*, only a length of InLen data are available for training the gradient boost trees model to be tested in the following out-of-sample. Please note that, rather than fix the length of in-sample window, the starting date of all in-sample windows are fixed and proceeding in a step of out-of-sample windows length OutLen. In step 3, it measures the performance of the whole out-of-sample.

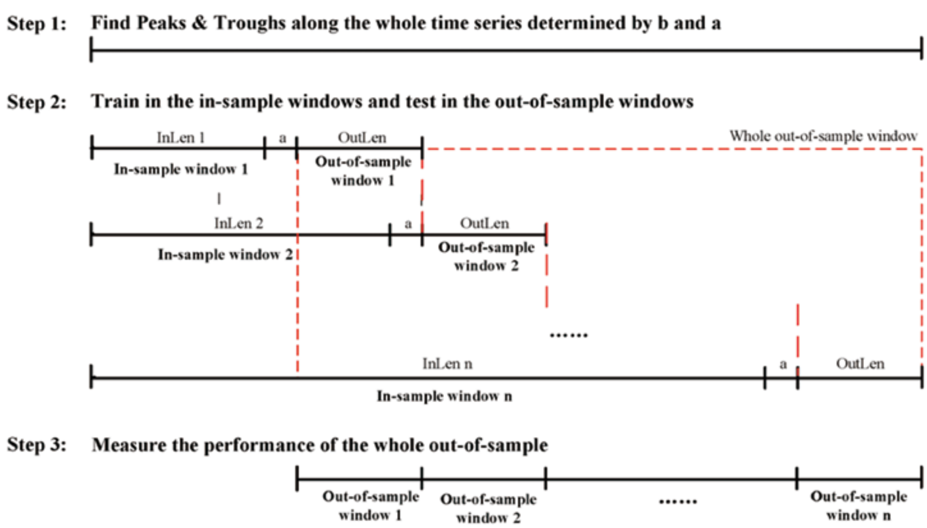


Figure 3
Diagrammatic Classification Process Within Rolling Windows

It first trains the data in the in-sample window [2005.04.08, 2007.04.11] and tests the classification-based trading strategy in the out-of-sample window [2007.04.12, 2007.10.12]. The implementation has been done 17 times for various in-sample windows sliding in a step of OutLen=131 trading days within [2007.10.12, 2016.06.30]. Note that the PnL derived from the whole out-of-sample windows can be achieved by the individual investor if he/she enters the market on April 12, 2007.

Through a number of tests employing the gradient boosted classification trees models, the results show that the single feature *PI* always outperforms the result of *TI* and the combination of them. Figure 4 thus investigates the performances of the whole out-of-sample with a single feature *PI* as a function of the starting date. It shows that, the shorter the out-of-sample window, the larger the Sharpe Ratio and PnL/MD. Although the values drop around the starting date of April 24, 2015, mainly due to the crash regime in the out-of-sample (as shown in the top panel of Figure 2), the Maximum Drawdown exhibits approximate plateaus of the value as a function of the starting date, which gives confidence in the reliability of the feature *PI* and the gradient boosted classification trees models.

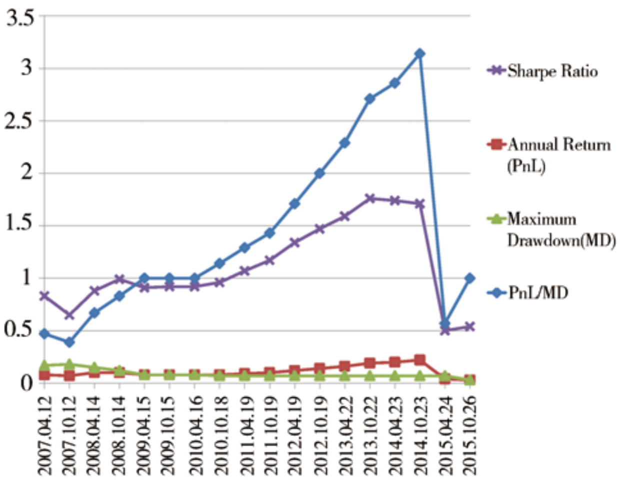


Figure 4
Performance of the Gradient Boosted Classification Trees Models With a Single Feature PI as a Function of the Starting Date

In more detail, when the investor enters the market on October 22, 2013, Figure 5 specifically shows a boosted classification tree of CSI 300 trained in the in-sample window [2005.04.08, 2013.10.21], preparing for the test in the out-of-sample window [2013.10.22, 2014.04.23]. The fuchsia circle represents the root node, the green nodes represent the intermediate nodes, the red terminal leaf nodes with positive score indicate the "buy" signal, and the blue nodes with negative values represent the "sell" signal. Thus, if $PI(t_i)<0.041985$, it gives the "sell" signal, if $PI(t_i)\geq 0.041985$, it presents the "buy" signal.

Table 2 presents the corresponding trading strategy and performance of the whole out-of-sample window [2013.10.22, 2016.06.30]. As the trading signals and the PnL shown in the top panel, the neat "buy-and-hold" strategies derived from the peak indicator *PI* successfully prevent the loss from the market volatility, as well as the bubble that crashed in June 2015.

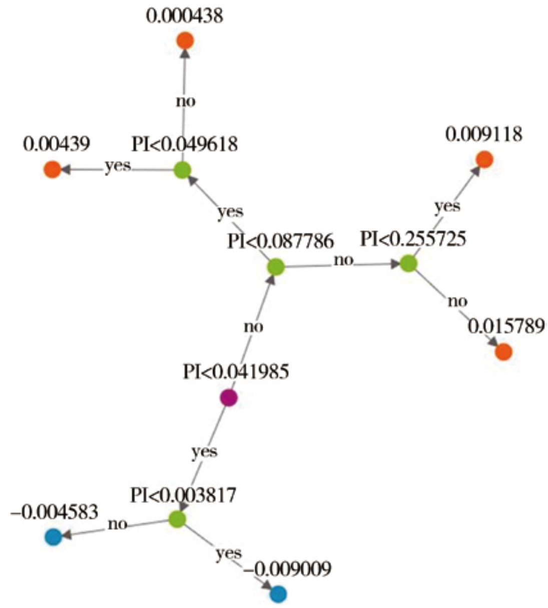
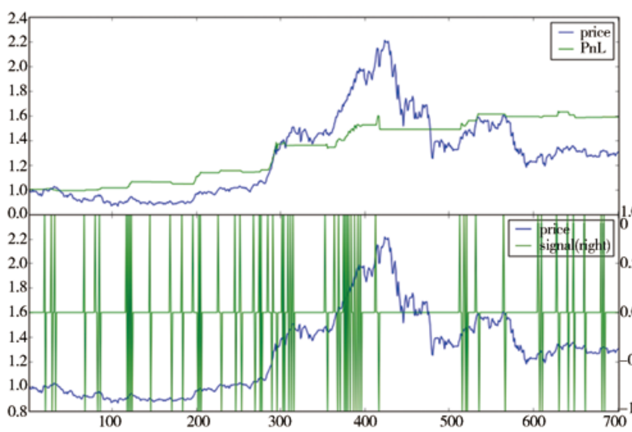


Figure 5
A Boosted Classification Tree of CSI 300 Trained in the In-Sample Window [2005.04.08, 2013.10.21]

Table 2
Performance of CSI 300 in the Whole Out-of-Sample Window [2013.10.22, 2016.06.30]

Trading strategy and the PnL



Statistical measures	Values
Sharpe Ratio	1.76
Annual Return (PnL)	0.19
Maximum Drawdown (MD)	0.07
PnL/MD	2.71
Trading days	703.00
Trade counts	54.00

CONCLUSION

In this paper, based on the gradient boosted classification trees models, a novel implementation of trading rule discovery associated with the convexity-concavity indicators has been proposed on the CSI 300. The performance with the buy-and-hold trading rule provides a further insight into the nature in recognizing the regime change of the Chinese stock market.

It is very remarkable to see the peak indicator PI can be regarded as an effective feature for developing automated trading strategy, which suggests an obvious pattern of upward convexity within the historical time series of the CSI 300. Looking at the performance of the trading strategy derived from this convexity indicator in recent times, it shows that the technical trading rule in general have a better short term predictive ability than that of long term, in terms of the values of Sharpe Ratio and PnL/MD.

The gradient boosting machine also demonstrates the possibility in improving the predictive capacity and lessening the Maximum Drawdown (MD). It thus would be more exciting to combine all these techniques to make an end-to-end system that scales to even larger real data and more complex trading rules in the future.

REFERENCES

Allen, F., & Karjalainen, R. (1999). Using genetic algorithms to find technical trading rules. *Journal of Financial Economics, 51,* 245-271.

Booth, A., Gerding, E., & McGroarty, F. (2014). Automated trading with performance weighted random forests and seasonality. *Expert Systems With Applications, 41,* 3651-3661.

Cervello-Royo, R., Guijarro, F., & Michniuk, K. (2015). Stock market trading rule based on pattern recognition and technical analysis: Forecasting the DJIA index with intraday data. *Expert Systems With Applications, 42,* 5963-5975.

Chen, T., & Guestrin, C. (2016). *Xgboost: A scalable tree boosting system.* arXiv:1603.02754.

Friesen, G. C., Weller, P. A., & Dunham, L. M. (2009). Price trends and patterns in technical analysis: A theoretical and empirical examination. *Journal of Banking & Finance, 33,* 1089-1100.

Hu, Y., Liu, K., Zhang, X., Su, L., Ngai, E., & Liu, M. (2015). Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review. *Applied Soft Computing, 36,* 534-551.

Johansen, A., Ledoit, O., & Sornette, D. (2000). Crashes as critical points. *International Journal of Theoretical and Applied Finance, 3,* 219-255.

Johansen, A., & Sornette, D. (1999). Predicting financial crashes using discrete scale invariance. *Journal of Risk, 1,* 5-32.

Kwon, Y.-K., & Moon, B.-R. (2007). A hybrid neurogenetic approach for stock forecasting. *Neural Networks, IEEE Transactions, 18,* 851-864.

Lacasa, L., Luque, B., Ballesteros, F., Luque, J., & Nuno, J. C. (2008). From time series to complex networks: The visibility graph. *Proceedings of the National Academy of Sciences, 105,* 4972-4975.

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

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management, 45,* 427-437.

Sornette, D. (2009). *Why stock markets crash: Critical events in complex financial systems*. Princeton University Press.

Wen, Q., Yang, Z., Song, Y., & Jia, P. (2010). Automatic stock decision support system based on box theory and SVM algorithm. *Expert Systems With Applications, 37,* 1015-1022.

Yan, W., & Serooskerken, E. V. T. V. (2015). Forecasting financial extremes: A network degree measure of super-exponential growth. *PLoS One, 10*(9).