Applied Soft Computing 13 (2013) 806–816

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

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

Integrating piecewise linear representation and weighted support vector machine for stock trading signal prediction

Linkai Luo*, Xi Chen

Department of Automation, Xiamen University, Xiamen 361005, PR China

ARTICLE INFO

Article history:
Received 5 February 2012
Received in revised form 14 October 2012
Accepted 27 October 2012
Available online 10 November 2012

Keywords:
Stock trading signal prediction
PLR
SVM

ABSTRACT

Piecewise linear representation (PLR) and back-propagation artificial neural network (BPN) have been integrated for the stock trading signal prediction recently (PLR–BPN). However, there are some disadvantages in avoiding over-fitting, trapping in local minimum and choosing the threshold of the trading decision. Since support vector machine (SVM) has a good way to avoid over-fitting and trapping in local minimum, we integrate PLR and weighted SVM (WSVM) to forecast the stock trading signals (PLR–WSVM). The new characteristics of PLR–WSVM are as follows: (1) the turning points obtained from PLR are set by different weights according to the change rate of the closing price between the current turning point and the next one, in which the weight reflects the relative importance of each turning point; (2) the prediction of stock trading signal is formulated as a weighted four-class classification problem, in which it does not need to determine the threshold of trading decision; (3) WSVM is used to model the relationship between the trading signal and the input variables, which improves the generalization performance of prediction model; (4) the history dataset is divided into some overlapping training–testing sets rather than training–validation–testing, which not only makes use of data fully but also reduces the time variability of data; and (5) some new technical indicators representing investors' sentiment are added to the input variables, which improves the prediction performance. The comparative experiments among PLR–WSVM, PLR–BPN and buy-and-hold strategy (BHS) on 20 shares from Shanghai Stock Exchange in China show that the prediction accuracy and profitability of PLR–WSVM are all the best, which indicates PLR–WSVM is effective and can be used in the stock trading signal prediction.

1. Introduction

In the stock market, there are two inherent characteristics: nonlinearity and time variability. The nonlinearity implies that the relationship between the stock price and its affecting factors is highly nonlinear while the time variability indicates the relationship changes over time. These two characteristics pose a great challenge on stock forecasting.

Recently, there are two interesting researches on stock forecasting [1]. One focuses on the prediction of the price variation in short-term. It will meet the challenge from the high frequency characteristic of data. The other one focuses on the prediction for the turning points of price [1–3]. Commonly the turning points have a longer period than the price variation in short-term, so the high frequency characteristic of data can be reduced. In addition, the turning point is more important than the non-turning point because it can supply more profits if it is predicted accurately. This paper focuses on the prediction for the turning points of stock price movement.

The first issue in the prediction of the turning points is to determine the turning points. There are many methods to extract valuable information from a time sequence, such as piecewise linear representation (PLR) [1,2,4–6], discrete Fourier transforms [7], wavelets [8], and symbolic mapping [9–11]. Among these methods, PLR may be the most suitable one for the checking of turning points because the joint points between adjacent segments generated by PLR just indicate the change of the trends.

The next issue is modeling the relationship between the stock price variation and the impact factors. Artificial neural networks (ANNs) have been widely used in this area. Zhang and Zhou [12] discussed some problems involved in applying ANN to stock prediction, such as the optimal length of training data, the selection of the network inputs, and so on. Kwon and Moon [13] applied recurrent neural network to predict the change rate of closing price and showed a notable improvement over buy-and-hold strategy (BHS) on the test for 36 companies in NYSE and NASDAG from 1992 to 2004. Chang et al. [1,2] made use of back-propagation artificial neural network (BPN) to model the relationship between the stock trading signals and the impact factors. However, ANNs

* Corresponding author.
E-mail address: luolk@xmu.edu.cn (L. Luo).

1568-4946/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.10.026

exists some disadvantages in avoiding over-fitting and trapping in local minimum. Cao and Tay [14] applied support vector machine (SVM) with additive parameters to predict the stock price, in which the regularized parameter determining the tradeoff between the empirical risk and confident risk as well as the tube size in each training point are adjusted by the distant to recent training data point. The goal of Cao and Tay is to predict the price accurately rather than the turning point. In addition, the performance criteria in Cao and Tay are normalized mean squared error, mean absolute error, and directional symmetry rather than profitability. Generic algorithm (GA) is often used in modeling process [1,2,13,15], where the function of GA is to select the model's parameters. For example, Chang et al. [1,2] apply GA to find the best threshold of PLR, while GA is used to find the best weights and topologies of ANNs in [13,15].

Chang et al. [1] integrated PLR and BPN (PLR–BPN) to check the turning points where PLR is used to generate the turning points from the history data while BPN is used to construct the regression model between the trading signals and the technique indexes. Chang et al. [1,2] have shown the effectiveness of PLR–BPN. However, there are some disadvantages in PLR–BPN. Firstly, the model generated by PLR–BPN does not been updated during the whole prediction term from 2005/10/01 to 2006/04/12. Considering the inherent time-varying characteristic of stock data, it may not be the best scheme keeping the prediction model unchanged during a long prediction term. Secondly, BPN has some disadvantages in avoiding over-fitting and trapping in local minimum. Thirdly, the threshold of the trading decision is difficult to determine. Chang et al. [2] propose a dynamic threshold method to overcome this disadvantage, in which the threshold is transferred to a discount factor in exponential smoothing (ES) and the bound parameters for the output of ES. However, setting these parameters is also uneasy. In addition, the importance of each turning points is same in [1,2]. It may be more reasonable that the importance of each turning point is different. For example, the turning point with a big turning trend should be more important than the one with a small turning trend.

To overcome the disadvantages of PLR–BPN, this paper integrates PLR and weighted SVM (WSVM) to predict the turning points of stock price movement where the function of PLR is same as in PLR–BPN while WSVM is used to model the relationship between the turning points and the impact factors. For simplicity, we notate it as PLR–WSVM. The main reason we choose SVM is the excellent generalization ability as well as all solutions of SVM model are globally optimal [16–19]. The second reason is that it has shown a good performance in solving the problems with small sample size [14,16,18,20]. It is well-known that the relationship between the turning points of stock price series and the impact variables is inherently time-varying. Up to now, there has not been a popular method to track time-varying data in machine learning society. Partitioning the whole dataset into some relatively small intervals is an ordinary approach, in which the prediction model in each small interval can be viewed approximately as non time-varying. Because the sample size in each small interval is small, SVM may be a better modeling tool. The last one is that WSVM can efficiently process the weights of each instance.

Except for making use of WSVM in modeling, PLR–WSVM also provides some other features different from Chang et al. [1,2] in the partition of history data, the setting of the weight to each training instance and the selection of input variables.

The rest of this paper is organized as follows. Section 2 briefly introduces PLR and SVM technology. In Section 3, the framework of PLR–WSVM is presented. The history data are firstly divided into some relatively small overlapping training–testing sets so that the data in each training–testing sets can be approximately viewed as non time-varying. Then, PLR is used to generate the turning points of each training set. Furthermore, the weights of the turning points are set by the change rate of the closing price between the current turning point and the next one. After some new technical indicators representing investors' sentiment are added to input variables, a four-class weighted classification problem is formulated and WSVM is used to model it. We also provide two common investment strategies to investigate the profitability of PLR–WSVM. In Section 4, we carry out some comparison experiments on 20 shares among PLR–WSVM, PLR–BPN and BHS to illustrate the performance of PLR–WSVM. Finally, the conclusions are given in Section 5.

2. Backgrounds

2.1. PLR

Given a time series $T = \{y_1, y_2, \ldots, y_l\}$, the PLR of T means the piecewise approximation straight lines, which can be described by

$$T_{PLR} = \{L_1(y_1, \ldots, y_{t_1}), L_2(y_{t_1+1}, y_{t_1+2}, \ldots, y_{t_2}), \ldots, \\ L_k(y_{t_{k-1}+1}, y_{t_{k-1}+2}, \ldots, y_l)\},$$

where t_i is the end time of the ith segment, $L_i(y_{t_{i-1}+1}, y_{t_{i-1}+2}, \ldots, y_{t_i})$ $(1 \le i \le k)$ indicates the approximation straight line to $y_{t_{i-1}+1}, y_{t_{i-1}+2}, \ldots, y_{t_i}$. Because t_i indicates a change of the movement trends, t_i is often named as turning point.

The approximation problem with PLR can be framed in several ways [4]:

- Given a time series T, produce the best representation using only k segments.
- Given a time series T, produce the best representation such that the maximum error for any segment does not exceed some user-specified threshold δ.
- Given a time series T, produce the best representation such that the combined error of all segments is less than some user-specified threshold.

There are two classical ways to find the approximation line. One is linear interpolation while the other one is linear regression. The algorithms for PLR can be divided into following three types [4]:

- *Sliding windows*: A segment is grown until it exceeds some error bound. The process repeats with the next data point not included in the newly approximated segment.
- *Top-down*: The time series is recursively partitioned until some stopping criteria is met.
- *Bottom-up*: Starting from the finest possible approximation, segments are merged until some stopping criteria are met.

In this paper, Top-down algorithm is used to segment the stock data and the linear interpolation is used to generate the approximation line. The object of segment is producing the best representation under the condition that the maximum error for any segment does not exceed the given threshold δ.

The result of PLR is greatly impacted by the threshold δ. There are only a few of turning points if δ is large. Contrarily, there are many turning points when the threshold value δ is small. Fig. 1 shows the results of PLR with δ = 0.1, 0.15 and 0.2 on a time-series. From Fig. 1, we can see that the turning points are greatly impacted by δ. Therefore, it is important to choose a suitable δ for individual shares.

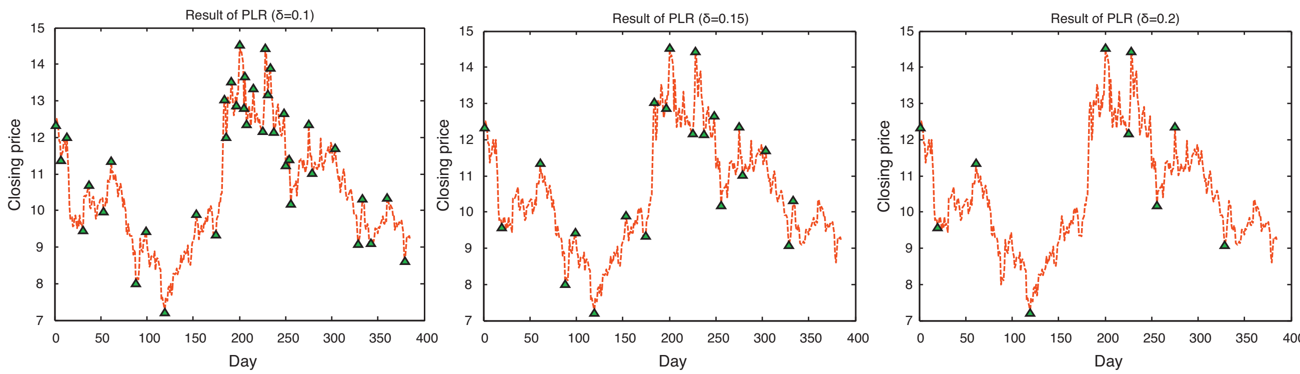

Fig. 1. The turning points obtained by PLR with $\delta = 0.1, 0.15, 0.2$.

2.2. SVM

The main idea of SVM proposed by Vapnik [16] is to generate a classification hyper-plane that separates two classes of data with the maximum margin. The standard SVM model is as follows:

$$\begin{aligned} \min_{w,b,\xi_i} \quad & \tfrac{1}{2}||w||^2 + C\sum_{i=1}^{l}\xi_i \\ \text{s.t.} \quad & y_i(\langle w, \phi(x_i)\rangle + b) + \xi_i \geq 1, \quad i = 1, 2, \ldots, l \\ & \xi_i \geq 0, \quad i = 1, 2, \ldots, l, \end{aligned} \tag{1}$$

where $x_i \in R^n$ and $y_i \in \{-1, +1\}$ are respectively the training instance and the corresponding class label, ϕ is a nonlinear map from the original space to a high dimensional feature space, w is the normal vector of hyper-plane in the feature space, b is a bias value, $\langle\cdot, \cdot\rangle$ denotes the inner product of two vectors, ξ_i ($i = 1, \ldots, l$) are slack variables, and C is a predefined parameter that balances the training accuracy and generalization ability. The dual model of (1) is

$$\begin{aligned} \min_{\alpha} \quad & \tfrac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j k(x_i, x_j) - \sum_{i=1}^{l}\alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^{l} y_i \alpha_i = 0, \\ & 0 \leq \alpha_i \leq C, \quad i = 1, 2, \ldots, l, \end{aligned} \tag{2}$$

where $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j)\rangle$ is named as kernel function. The decision function of the binary classification problem can be obtained by

$$y(x) = sign(f(x)) = sign\left(\sum_{x_i \in SV} y_i \alpha_i^* k(x, x_i) + b^*\right), \tag{3}$$

where α^* is an optimal solution of convex programming problem (2), SV is the set of support vectors, $b^* = y_j - \sum_{x_i \in SV} y_i \alpha_i^* k(x_i, x_j)$, if $0 < \alpha_j^* < C$.

The commonly used kernel functions in SVM are linear function $k(x_i, x_j) = x_i^T x_j$, Gaussian radial basis function (GRBF) $k(x_i, x_j) = e^{-g||x_i - x_j||^2}$ and polynomial function $k(x_i, x_j) = (g x_i^T x_j + a)^d$.

The standard SVM is extended to weighted SVM (WSVM) [21] or fuzzy SVM [22] when each training instance has different weights. In WSVM, the models (1) and (2) are respectively transformed to

$$\begin{aligned} \min_{w,b,\xi_i} \quad & \tfrac{1}{2}||w||^2 + C\sum_{i=1}^{l}\mu_i\xi_i \\ \text{s.t.} \quad & y_i(\langle w, \phi(x_i)\rangle + b) + \xi_i \geq 1, \quad i = 1, 2, \ldots, l, \\ & \xi_i \geq 0, \quad i = 1, 2, \ldots, l, \end{aligned} \tag{4}$$

and

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j k(x_i, x_j) - \sum_{i=1}^{l}\alpha_i, \\ \text{s.t.} \quad & \sum_{i=1}^{l} y_i \alpha_i = 0, \\ & 0 \leq \alpha_i \leq C\mu_i, \quad i = 1, 2, \ldots, l, \end{aligned} \tag{5}$$

where μ_i ($i = 1, \ldots, l$) indicates the weight of instance x_i. The decision function for WSVM is the same as the standard SVM.

SVM can be extended to the k-class problem ($k > 2$) although SVM is originally designed to two-class classification problem. The common way for k-class problem is to decompose it into a series of two-class problems. There are mainly two decomposition methods: one-versus-one (1-v-1) and one-versus-rest (1-v-r) [23]. In 1-v-1 method, one must construct $k(k-1)/2$ binary classifiers f_{ij}($i = 1, \ldots, k$, $j = i + 1, \ldots, k$) that separates class i and class j. The decision rule for the k-class problem is obtained by majority voting method, i.e.

$$y(x) = arg\max_{1 \leq i \leq k} \ v_i(x), \tag{6}$$

where $v_i(x)$ is the votes for class i.

Alternatively, in 1-v-r method, one must construct k binary classifiers f_i($i = 1, \ldots, k$) that separates class i and the rest $k - 1$ classes. And the decision rule for the k-class problem is

$$y(x) = arg\max_{1 \leq i \leq k} \ f_i(x). \tag{7}$$

3. The PLR–WSVM for stock trading signal detection

In this section, we will describe the proposed method PLR–WSVM. To reduce the time-varying characteristic of stock transaction data, the whole history dataset is firstly divided into some overlapping training–testing sets. Then, PLR is used to generate turning points of each training set and the weights of the turning points are also computed. After some new technical indicators are added to the input variables, a four-class weighted classification problem is formulated and WSVM is used to model it. Finally, the profits of two common investment strategies are computed to demonstrate the effect of PLR–WSVM.

3.1. The partition of history data

In time-series data analysis, the whole dataset is often divided to some overlapping training–validation–testing sets [13,14]. There are two reasons for doing this. One is that the time-varying feature of data can be reduced while the other is that the order of time can be maintained. Considering the k-fold cross validation on training

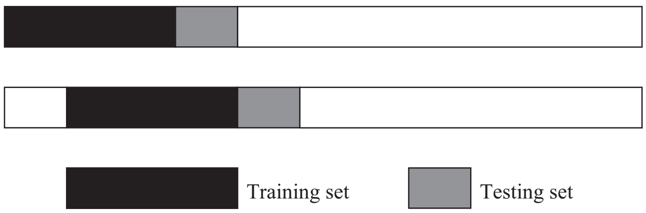

Fig. 2. A example for two successive training–testing sets in overlapping partition.

set is suitable for big size or small size of dataset while the validation with independent dataset is only suitable for big one, we divide the whole dataset to some overlapping training–testing sets. Because the independent dataset is removed, the data is used more fully and the time-variability of the data is also reduced further. Fig. 2 displays an example for two successive training–testing sets in overlapping partition. Suppose the size of the whole dataset is r, while the size of each training set and testing set are r_1 and r_2 respectively. Then, the whole dataset will be divided into q overlapping training–testing sets, in which

$$q = \left\lceil \frac{r - r_1}{r_2} \right\rceil, \tag{8}$$

where $\lceil x \rceil$ denotes the minimal positive integer that is not less than x. Here, r_2 can be viewed as the updating period for the prediction model, i.e., the prediction model established after r_2 days must be retrained with new data.

3.2. Generating turning points by PLR

Top-down method is used to generate the turning points and the pseudo-code is similar to that of Chang et al. [1]. However, there are three differences between our method and [1].

The first one is the standardization of price. Because the threshold δ represents the maximum absolute error between the approximation line and the original price sequence, the range of threshold δ in PLR is affected by the price level. For example, the range of the threshold for price level 100 and 1 should be different. In stock market, the prices may vary widely for different shares or the same share at different time. To obtain a same range of the threshold for different price levels, the standardization of price is needed. In this study, the price is standardized to [0,1] according to

$$\tilde{p}(t) = \frac{p(t) - m}{M - m}, \tag{9}$$

where $p(t)$ and $\tilde{p}(t)$ are original and scaled price respectively, M and m are the maximum and minimum of original prices respectively. After standardization of price, the range of threshold δ for different price levels can be set with the same.

The second one is the classification of trading signals. The trading signals obtained by PLR in our method are divided into four classes, instead of the real value used in Chang et al. [1]. The turning points with trough or peak are labeled as strongly recommended buying point (SBP) or strongly recommended selling point (SSP) while the other points are labeled as ordinarily recommended buying point (OBP) or ordinarily recommended selling point (OSP) by the up or down of the closing price in the next day. The trading decision is only executed on SBP and SSP. Through dividing the trading signals into four classes, we do not need to choose the threshold of trading decision.

The third one is the weight of trading signal. The trading signals generated by PLR in our method are set by different weights while they are same in Chang et al. [1]. Generally, the weights of trading signals should be different. For example, the weight of SBP or SSP should be bigger than that of OBP or OSP because it can provide more profit. It may be reasonable to set the weight of a SBP or SSP with the change rate of closing prices between the current tuning point and the next one. Therefore, the weight of a SBP or SSP is set according to

$$\mu(s_t) = \frac{|p(ns_t) - p(s_t)|}{p(s_t)}, \tag{10}$$

where s_t and ns_t respectively indicate the current and next turning point. The weights for OBPs and OSPs are set with the same by

$$\mu(o_t) = \lambda * \min_{s_t} w(s_t), \tag{11}$$

where o_t indicates OBP or OSP, $0 < \lambda \le 1$ is a scaled factor. Finally, the weight is normalized by

$$\mu(t) = \frac{\mu(t)}{\sum_i \mu(i)}. \tag{12}$$

3.3. Input variable selection

The input variables are the factors that impact on trading signal. In the stock prediction problem, the selection of input variables is very important. The common input variables are some technical indicators, such as moving average (MA), relative strength index (RSI), and transaction volume (TV). Many scholars have proposed a lot of technical indicators to predict the trading signal [24]. However, it is still not clear that which combination of technical indicators can predict the trading signal well.

Inspired by empiricism, we add some new technical indicators representing investors' sentiment to input variables. The first added sentiment indicator is the average transaction price (ATP). ATP is defined as ATP = TM/TV where TM indicates the transaction money in one day. Compared with closing price, ATP more truly reflects the average cost of transaction and can provide more useful information. Conversely, closing price is easier to manipulate so that it often gives false information. In addition, considering the change rate of ATP relative to the previous trading day can provide additional useful information, we also add it to the input variables.

The second is the amplitude of the price movement (ALT) in one day, which is defined as

$$\mathrm{ALT} = \frac{p_h(t) - p_l(t)}{p_l(t)}$$

where $p_h(t)$ and $p_l(t)$ are the highest price and the lowest price respectively on the tth day. ALT reflects the activity of stock, which is important to investors. If ALT of a stock is zero, it certainly cannot bring any profit to investors.

The third added sentiment indicator is the index for the type of K-line (ITL) which is defined as

$$\mathrm{ITL} = \begin{cases} 1 & \text{if } p_c(t) > p_o(t) \\ -1 & \text{otherwise} \end{cases},$$

where $p_c(t)$ and $p_o(t)$ are the closing price and the opening price respectively on the tth day. ITL plays a significant impact on the investors' sentiment. Generally, ITL = 1 gives a positive impact to buy order while ITL = −1 has a negative impact.

Considering the price movement of individual share is generally related to the market movement, we also add some market indexes to the input variables of individual share. The used technical indicators in this paper are listed in Table 1.

To avoid variables in greater numeric ranges dominating those in smaller numeric ranges and numerical difficulties during calculation [25], the selected input variables are normalized by

$$\tilde{x}_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \tag{13}$$

Table 1
the technical indicators used as input variable. $p_c(t)$, $p_o(t)$, $p_h(t)$, $p_l(t)$, $\bar{p}(t)$, $m(t)$, respectively indicate the closing price, the opening price, the highest price, the lowest price, ATP, TM on the tth day.

Technical indicators	Explanation
The closing price	$p_c(t)$
The opening price	$p_o(t)$
The highest price	$p_h(t)$
The lowest price	$p_l(t)$
The average transaction price (ATP)	$\bar{p}(t)$
The amplitude of the price movement (ALT)	$(p_h(t)-p_l(t))/p_l(t)$ (the ALT for composite index is also included)
The index for the type of K-line (ITL)	The type of K-line. If $p_c(t)>p_o(t)$, ITL = 1, otherwise ITL = −1 (the ITL for composite index is also included)
The change rate of average transaction price to the previous trading day	$(\bar{p}(t)-\bar{p}(t-1))/\bar{p}(t-1)$ (the change rate of closing price for composite index is also included)
The change rate of transaction money to previous trading day	$(m(t)-m(t-1))/m(t-1)$. (The one for composite index is also included)
The difference of MA between the short run and the long run (DMA)	10 MA–50 MA, where 10 MA and 50 MA are the moving average of closing price for 10 days and 50 days respectively
The average of DMA (AMA)	The average of DMA for 10 days

where x_i is the original data, $x_{\max}$ and $x_{\min}$ mean the maximum and the minimum values.

3.4. Constructing prediction model by WSVM

After generating trading signals by PLR and selecting input variables, a four-class weighted classification problem is constructed for each overlapping training–testing

$$T^{(i)} = T^{(i,tr)} \cup T^{(i,ts)}, \quad i = 1, 2, \ldots, q,$$

where

$$T^{(i,tr)} = \{(x_t^{(i,tr)}, y_t^{(i,tr)}, \mu_t^{(i,tr)}) | x_t^{(i,tr)} \in R^n, \quad y_t^{(i,tr)} \in \{1, 2, 3, 4\},$$
$$\mu_t^{(i,tr)} \geq 0, \quad t = 1, 2, \ldots, r_1\},$$

and

$$T^{(i,ts)} = \{(x_t^{(i,ts)}, y_t^{(i,ts)}) | x_t^{(i,ts)} \in R^n, \ y_t^{(i,ts)} \in \{1, 2, 3, 4, \},$$
$$t = 1, 2, \ldots, r_2\},$$

respectively denote the training set and testing set, $x_t^{(i,tr)}$ or $x_t^{(i,ts)}$ is the instance, $y_t^{(i,tr)}$ or $y_t^{(i,ts)}$ is the corresponding class label: label '1', '2', '3' and '4', respectively indicates SBP, OBP, OSP, SSP, $\mu_t^{(i,tr)}$ is the weight of the training instance which is computed according to (10)–(12).

The WSVM is used to model this four-class problem. Usually the number of instances for SBP or SSP is less than OBP or OSP. Hence there exists class imbalance problem. Considering that 1-v-r method will increase the class imbalance although there are less binary classification problems, we use 1-v-1 method to decompose the four-class problem into six binary classification problems.

In each binary classification problem, the parameters of WSVM model is chosen by the grid search based on k-fold cross validation. The model' parameters include C and kernel parameters. For example, they are C and g if the kernel function is GRBF. It is unknown which pair of parameters are best for a given problem. Therefore, we must carry out the parameter selection before training process. At present, the most commonly used parameter selection method is the grid search based on k-fold cross validation. In k-fold cross validation, the training set is first divided into k subsets of equal size, then one subset is tested using the classifier trained on the remaining $k-1$ subsets sequentially [25]. In grid search, the parameter space is first divided into a series of grids, in which a grid corresponds to a pair of parameters. Then an exhaustive search through a specified range is used to find the pair of parameters with the highest cross validation accuracy or the lowest testing error.

After the parameters are selected, the training set $T^{(i,tr)}$ is inputted to train WSVM model. Because WSVM model is a quadratic programming (QP) problem, training a WSVM model means the solution of a QP problem. The sequential minimal optimization (SMO) proposed by Platt [26] is a fast algorithm in the solution of large QP problem. The main two parts of SMO are the solution of two Lagrange multipliers and working set selection. Many scholars proposed some improvement rules in working set selection, see Refs. [27,28]. In this paper, we adopt the method of Fan et al. [28] to train WSVM model.

Once the model for four-class classification problem is trained, it will be used to find the trading signals on testing set $T^{(i,ts)}$.

4. Performance measure

The most important objective for stock prediction is the profit that it can make. It has been found that many rules do not make profits if the transaction costs are considered, although they seem to yield returns from the prediction accuracy [13,29–32]. To illustrate the prediction performance objectively, we bind two investment strategies to investigate the profitability of PLR–WSVM. The two investment strategies are demonstrated as follows, where b_s, b_m, v_m, c_b, c_s denote the balance number of share, balance money, total investment money, the transaction cost rates of purchasing and selling, respectively.

Strategy 1. If $y(t) = 1$, we always buy the share with one unit of money, i.e.,

$$b_s \leftarrow b_s + \frac{1}{p_c(t) \times (1 + c_b)}; \quad v_m \leftarrow v_m + 1 - b_m, \quad \text{if} \quad 0 \leq b_m \leq 1;$$
$$b_m \leftarrow \begin{cases} b_m - 1, & \text{if} \quad b_m > 1 \\ 0, & \text{if} \quad 0 \leq b_m \leq 1 \end{cases}, \tag{14}$$

If $y(t) = 4$ and $b_s > 0$, we sell all shares, i.e.,

$$b_m \leftarrow b_m + b_s \times p_c(t) \times (1 - c_s); \quad b_s \leftarrow 0. \tag{15}$$

Otherwise, no any action is executed.

Strategy 2. The total investment is limited in one unit of capital and the initial investment is set by one. If $y(t) = 1$ and $b_m > 0$, buy share with the balance money, i.e.,

$$b_s \leftarrow b_s + \frac{b_m}{p_c(t) \times (1 + c_b)}; \quad b_m \leftarrow 0. \tag{16}$$

If $y(t) = 4$ and $b_s > 0$, sell all shares and update b_m and b_s according to (15). Otherwise, no any action is executed.

The short-selling is not allowed in Strategy 1 and Strategy 2. In China stock market, most of the shares are not allowed to short-selling. Therefore, Strategies 1 and 2 are suitable for China stock market. Strategy 1 generally requires a large amount of investment while it is limited to one unit of capital for Strategy 2. Strategy 1 is suitable for the investors that possess a lot of funds while Strategy 2 is suitable for retail investors.

At the end of an investment period, all shares must be sold with the last day's closing price. The profit for Strategy 1 or 2 is computed by

$$p_m = \frac{b_m - v_m}{v_m}. \tag{17}$$

The framework of PLR–WSVM is illustrated by following algorithm PLR–WSVM.

Algorithm PLR–WSVM
Input: Dataset D and the parameters δ, r_1, r_2, λ, c_b, c_s
Output: The test accuracy, the profits, investments, the times for buying and selling

(1) Normalizing the dataset D according to (13).
(2) Compute the times q for training and testing according to (8).
(3) Set $i = 1$.
(4) While ($i \le q$).
  (a) Select the ith training set and testing set from D.
  (b) Generate the turning points from the ith training set and the ith testing set via PLR with the given threshold δ.
  (c) Set the weights of each instance in the ith training set according to (10)–(12).
  (d) Construct a four-class classification problem in which the classes 1–4 denote SBP, OBP, OSP, SSP respectively.
  (e) Train a four-class WSVM model from the ith training set by WSVM.
  (f) Apply the four-class WSVM model trained to compute the classification results on ith test set.
  (g) Set $i = i + 1$.
(5) Make a union of the test result of q times and compute the test accuracy.
(6) Compute the profits, investments, the times for buying and selling for Strategy 1 and Strategy 2 based on the test result.

Return

5. Numerical experiments

Twenty shares chosen randomly from Shanghai Stock Exchange are used to investigate the performance of PLR–WSVM. The time span for these shares is from 2010.1.4 to 2011.8.18. For each share, there are about 391 days of data. Twenty shares are divided into three types: downtrend, steady trend and uptrend. The market in this time span is in the downtrend. Hence the number of stock in the downtrend is more than that in the uptrend. Among twenty shares, nine are divided into the downtrend (Code: 600736, 600197, 600211, 600694, 600351, 600488, 600054, 600019, 600058), six are in the steady trend (Code: 600682, 600597, 600066, 600881, 600228, 600697), and other five are in the uptrend (Code: 600107, 600053, 600051, 600163, 600167). The rule to decide the type of a share is as follows: if the change rate of closing price between the staring point and the end of the test period for one share is lower than 10%, then it is divided into the downtrend; if the change rate is higher than 10%, then it is divided into the uptrend; otherwise, it is divided into the steady trend.

To fairly evaluate the performance of PLR–WSVM, we carry out some comparative experiments among PLR–WSVM, PLR–BPN and BHS. PLR–WSVM, PLR–BPN and BHS all run on the same training set and testing set. The shared parameters for PLR–WSVM, PLR–BPN and BHS are set as follows:

$$r_1 = 200, \quad r_2 = 20, \quad \delta = 0.15, \quad \lambda = 0.2, \quad c_b = 0.0035, \quad c_s = 0.0035.$$

LIBSVM-mat-2.89-1 [21] is adopted to solve WSVM model. The kernel function for SVM is GRBF and the grid search based on 5-fold cross validation is used to select the model's parameter. The Neural Network Toolbox in Matlab R2007b is used to construct BPN model. The parameters for BPN are set as follows: the number of hidden layer is 1, the transfer functions for the hidden layer and the output layer are $\text{tansig}(x) = 2/(1 + e^{-2x}) - 1$ and $\text{purelin}(x) = x$ respectively, the maximum learning times are 500, the number of the neurons for the hidden layer is set by 5-fold cross validation.

Tables 2–4 list the comparison results of PLR–WSVM, PLR–BPN and BHS on testing set for the shares in the downtrend, the steady trend and the uptrend, respectively (the transaction cost is also considered in the computation for the profit of BHS). In Tables 2–4, ACC means the accuracy, V_m represents the investment, N_b and N_s are the times of buying and selling, respectively. Figs. 3–5 show the predicted trading signals (i.e., $y(t) = 1$ or $y(t) = 4$) of PLR–WSVM and PLR–BPN on testing set for the shares in the uptrend, the steady trend and the downtrend, where 'Δ' and '$\triangledown$' denote the buying and selling signal, respectively.

From Table 2, we can see that PLR–WSVM achieves the best performance while BHS performs the worst on the shares in the downtrend. In the prediction accuracy of the four-class problem, PLR–WSVM wins PLR–BPN on seven shares, loses on two. And the average on nine shares for PLR–WSVM is 36.49%, which outperforms PLR–BPN with 4.37%. In the profit with Strategy 1, PLR–WSVM wins PLR–BPN on all nine shares, and the average for PLR–WSVM is 5.16% while it is −12.22% for PLR–BPN. In the profit with Strategy 2, PLR–WSVM also wins PLR–BPN on all nine shares, and the average for PLR–WSVM is 5.80% that is higher than PLR–BPN with 25.19%. The average profits with Strategies 1 and 2 for PLR–WSVM and PLR–BPN are 5.16%, 5.80%, −12.22% and −19.39%, respectively, all are better than −21.49% of BHS.

From Table 3, we can see the performances among PLR–WSVM, PLR–BPN and BHS on the shares in the steady trend are similar to that in the downtrend. In the prediction accuracy, PLR–WSVM wins PLR–BPN on four shares, loses on two. And the average for PLR–WSVM is 33.56% while it is 31.70% for PLR–BPN. In the profits with Strategies 1 and 2, they are 8.84% and 8.71% for PLR–WSVM, all are better than 4.63% and 3.14% of PLR–BPN. The profit of BHS is −3.46%, which is the lowest.

The result from Table 4 also shows PLR–WSVM achieves the best performance on the shares in the uptrend. However, PLR–BPN is inferior to BHS.

Compared with PLR–BPN and BHS, PLR–WSVM achieves the better profits on all three types of shares. However, there exist some differences. In the downtrend, the average profit with Strategy 1 for PLR–WSVM is higher than PLR–BPN and BHS by 17.38% and 26.65%, while it is 25.19% and 27.29% for Strategy 2. In the steady trend, PLR–WSVM is higher by 4.21% and 12.3% for Strategy 1, 5.57% and 12.17% for Strategy 2. In the uptrend, it is 9.29% and 1.64% for Strategy 1, 17.07% and 12.75% for Strategy 2. PLR–WSVM performs the best on the downtrend, near on the steady trend and the uptrend. Noticing that the market drops from 2010.1.4 to 2011.8.18, we think the market indexes may be more helpful for the prediction model of downtrend shares than that of steady and uptrend.

From Figs. 3–5, the number of the trading signals predicted by PLR–BPN is more than PLR–WSVM but the precision is lower than PLR–WSVM. On the average number of trading signals in the downtrend, steady trend and uptrend, they are 38.89, 52.33 and

Table 2
The comparison result on testing set among PLR–WSVM, PLR–BPN and BHS for the downtrend shares, in which the bold font indicates the better one between PLR–WSM and PLR–BPN.

Code of share	Method	ACC	Strategy 1				Strategy 2 ($V_m = 1$)			Profit for BHS
			V_m	Profit	N_b	N_s	Profit	N_b	N_s	
600736	PLR–BPN	32.29	9	−3.21	9	1	−14.80	1	1	−11.70
	PLR–WSVM	**36.98**	12	**29.11**	12	1	**27.02**	1	1	
600197	PLR–BPN	**34.90**	17	−10.38	20	2	−14.25	2	2	−24.05
	PLR–WSVM	28.65	1	**11.29**	2	2	**11.21**	2	2	
600211	PLR–BPN	33.33	13.52	−10.27	30	6	14.99	6	6	−13.20
	PLR–WSVM	**37.57**	3.74	**5.38**	13	5	**24.63**	5	5	
600694	PLR–BPN	29.84	3.36	−3.89	8	5	−24.25	5	5	−19.81
	PLR–WSVM	**37.17**	13.75	**17.84**	17	2	**8.07**	2	2	
600351	PLR–BPN	**33.16**	14	−21.01	14	1	−23.77	1	1	−34.94
	PLR–WSVM	32.64	13	**−5.87**	13	1	**−9.80**	1	1	
600488	PLR–BPN	33.88	8	−14.16	16	3	−25.06	3	3	−27.35
	PLR–WSVM	**35.52**	3	**−13.55**	3	1	**−15.32**	1	1	
600054	PLR–BPN	30.21	24.48	−13.15	39	3	−23.87	3	3	−16.03
	PLR–WSVM	**40.63**	2.95	**2.25**	6	3	**−2.40**	3	3	
600019	PLR–BPN	27.98	49.08	−15.11	57	6	−26.46	6	6	−31.84
	PLR–WSVM	**34.72**	7.97	**−11.28**	10	3	**−13.36**	3	3	
600058	PLR–BPN	33.51	11.30	−18.83	14	3	−37.05	3	3	−14.46
	PLR–WSVM	**44.50**	3	**11.23**	3	1	**22.16**	1	1	
Average	PLR–BPN	32.12	16.64	−12.22	23	3.33	−19.39	3.33	3.33	−21.49
	PLR–WSVM	**36.49**	6.71	**5.16**	8.78	2.11	**5.80**	2.11	2.11	

Table 3
The comparison result on testing set among PLR–WSVM, PLR–BPN and BHS for the steady trend shares, in which the bold font indicates the better one between PLR–WSM and PLR–BPN.

Code of share	Method	ACC	Strategy 1				Strategy 2 ($V_m = 1$)			Profit for BHS
			V_m	Profit	N_b	N_s	Profit	N_b	N_s	
600682	PLR–BPN	**32.09**	5	2.44	17	8	1.63	8	8	−4.05
	PLR–WSVM	31.55	2	**3.59**	3	2	**3.22**	2	2	
600597	PLR–BPN	35.11	16.68	−4.06	27	4	**8.35**	4	4	−3.82
	PLR–WSVM	**35.64**	13.02	**4.30**	16	2	−9.16	2	2	
600066	PLR–BPN	**33.68**	18.97	**13.08**	35	5	**−1.38**	5	5	−9.26
	PLR–WSVM	33.16	1	−3.88	2	2	−3.98	2	2	
600881	PLR–BPN	33.15	18	−6.39	38	7	1.59	7	7	−3.76
	PLR–WSVM	**33.16**	4.65	**0.34**	7	2	**4.43**	2	2	
600228	PLR–BPN	31.02	3.06	17.41	13	8	4.61	8	8	−1.75
	PLR–WSVM	**39.04**	4	**37.99**	8	2	**33.17**	2	2	
600697	PLR–BPN	25.13	5.85	5.29	18	7	4.01	7	7	1.90
	PLR–WSVM	**28.80**	4	**10.72**	9	5	**24.60**	5	5	
Average	PLR–BPN	31.70	11.26	4.63	24.67	6.50	3.14	6.50	6.50	−3.46
	PLR–WSVM	**33.56**	4.78	**8.84**	7.50	2.50	**8.71**	2.50	2.50	

Table 4
The comparison result on testing set among PLR–WSVM, PLR–BPN and BHS for the uptrend shares, in which the bold font indicates the better one between PLR–WSM and PLR–BPN.

Code of share	Method	ACC	Strategy 1				Strategy 2 ($V_m = 1$)			Profit for BHS
			V_m	Profit	N_b	N_s	Profit	N_b	N_s	
600107	PLR–BPN	38.22	6.10	1.94	20	8	−1.31	8	8	16.27
	PLR–WSVM	38.22	8	**15.33**	10	3	**13.95**	3	3	
600053	PLR–BPN	36.72	8.00	8.92	25	7	20.72	7	7	9.36
	PLR–WSVM	**38.42**	11	**24.76**	14	4	**44.03**	4	4	
600051	PLR–BPN	33.16	9.06	16.22	10	2	22.61	2	2	16.43
	PLR–WSVM	**36.32**	8.99	**21.78**	13	3	**50.17**	3	3	
600163	PLR–BPN	27.84	41.38	**6.17**	56	6	**11.13**	6	6	11.18
	PLR–WSVM	**35.23**	7	5.53	8	2	9.65	2	2	
600167	PLR–BPN	37.82	5	1.20	12	4	−2.03	4	4	19.47
	PLR–WSVM	**44.56**	1.91	**13.48**	5	4	**18.64**	4	4	
Average	PLR–BPN	34.75	13.91	6.89	24.60	5.40	10.22	5.40	5.40	14.54
	PLR–WSVM	**38.55**	7.38	**16.18**	10.00	3.20	**27.29**	3.20	3.20	

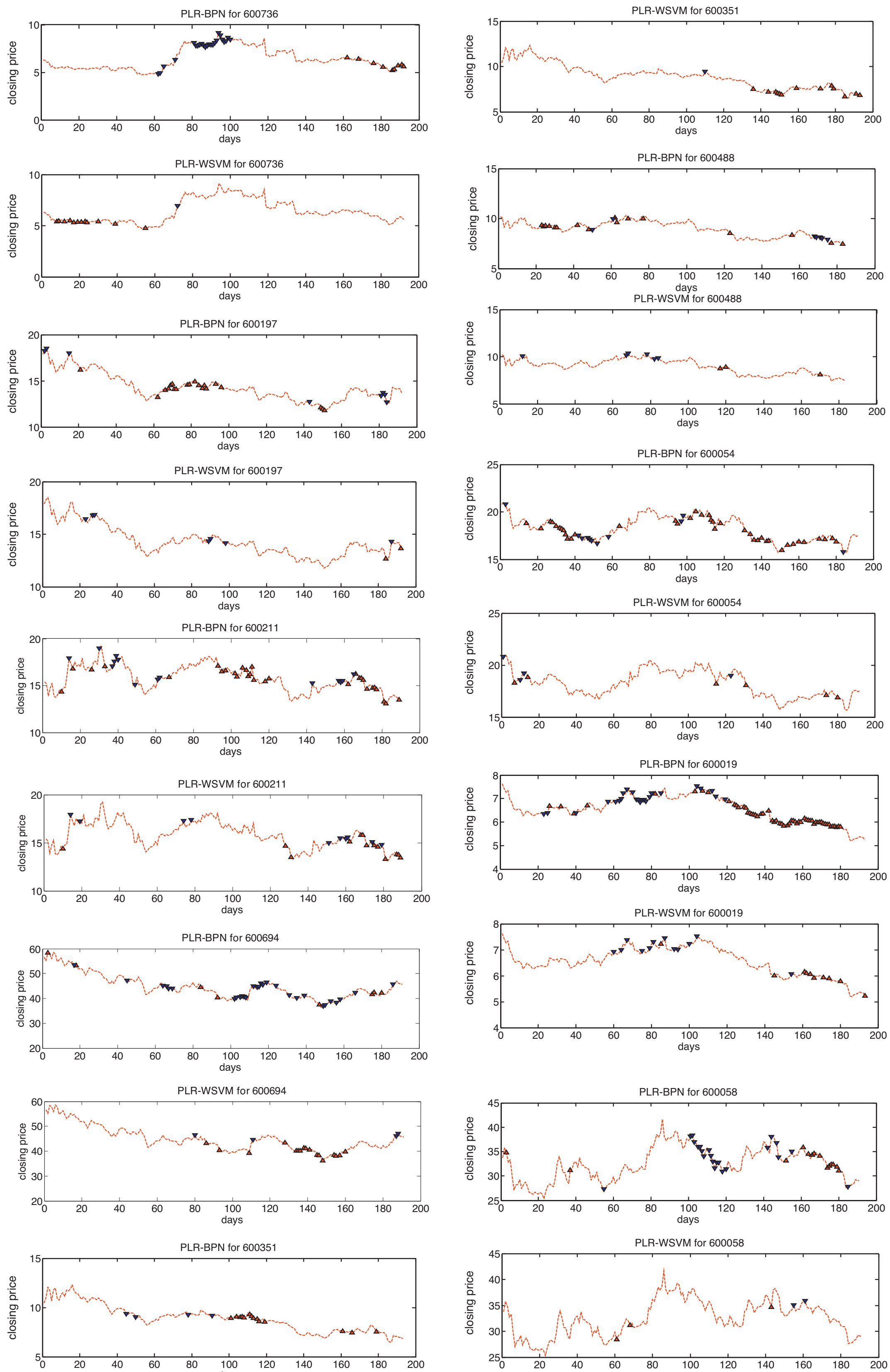

Fig. 3. The comparison of predicted trading signals between PLR–BPN and PLR–WSVM for the downtrend shares.

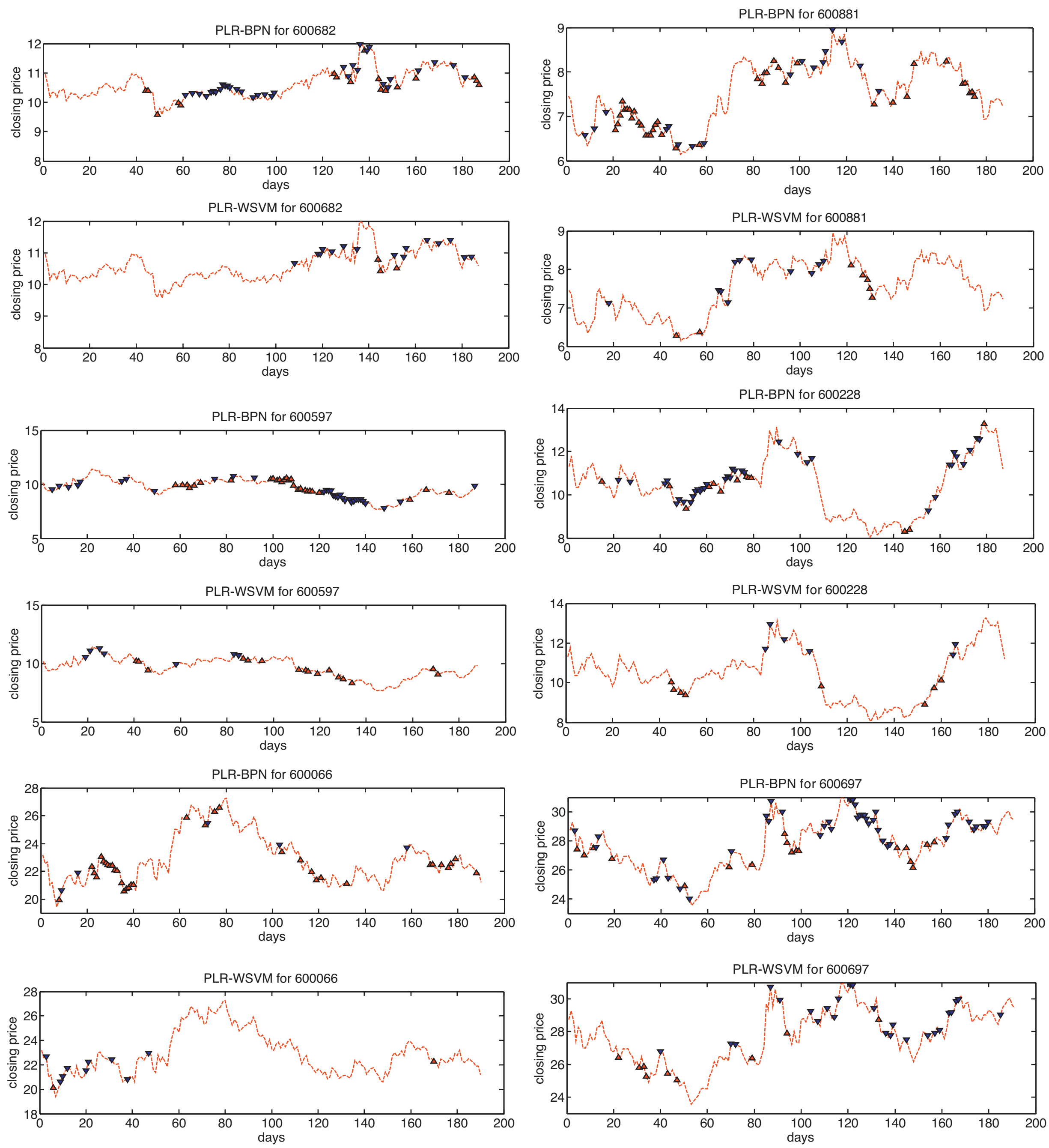

Fig. 4. The comparison of predicted trading signals between PLR–BPN and PLR–WSVM for the steady trend shares.

48.4 for PLR–BPN while they are 12.11, 19 and 18.6 for PLR–WSVM, respectively. On the average precision of trading signals in the downtrend, steady trend and uptrend, they are 18.30%, 21.01% and 21.09% for PLR–BPN while they are 34.30%, 24.48% and 22.19%, in which PLR–WSVM are higher than PLR–BPN with 16%, 3.47% and 1.1%, respectively. This is the main reason why PLR–WSVM performs better in making profit than PLR–BNP.

Figs. 3–5 also show both the trading signals predicted by PLR–WSVM and PLR–BPN are still far from the optimum. We can see the precision is not high. In addition, a lot of actual trading signals do not been recognized by PLR–WSVM and PLR–BPN. We think it is reasonable. The prediction of stock trend is a very difficult problem because of the great uncertainty and time-variability. Up to now, it is still luxurious to obtain a high accuracy or make a great profit. From the view of game theory, the high accuracy or great profit is doubtful.

The experiment results above show both PLR–WSVM and PLR–BPN can learn the valuable knowledge hidden in the history data, while PLR–WSVM performs better than PLR–BPN does. We think the first reason is the excellent generalization ability of SVM. Through structural risk minimum principle, SVM achieves a stronger generalization ability than BPN, which explains the fact that the prediction accuracy for PLR–WSVM is higher than PLR–BPN. Another reason is that all solutions in SVM are global

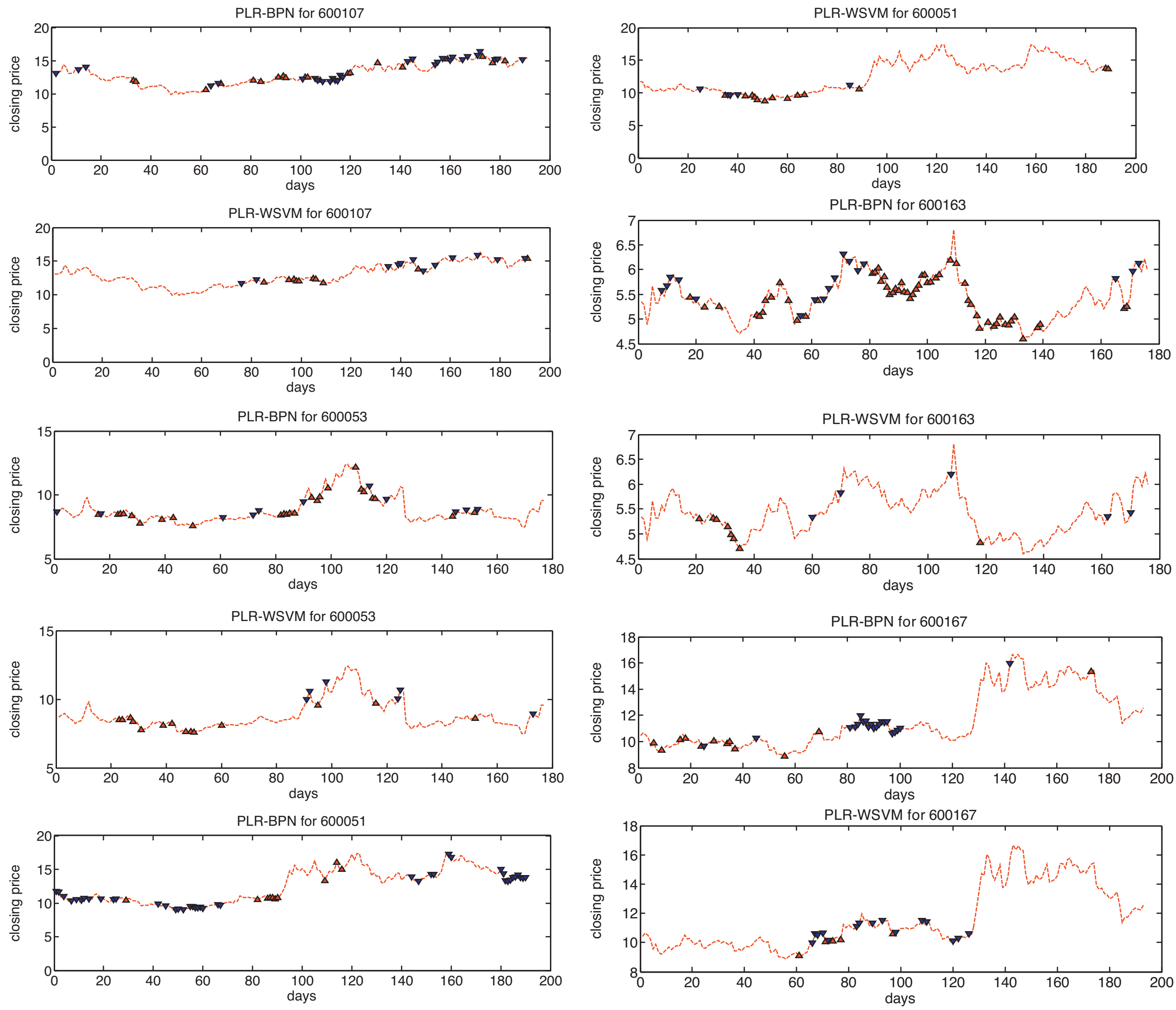

Fig. 5. The comparison of predicted trading signals between PLR–BPN and PLR–WSVM for the uptrend shares.

optimal while BPN often falls into local minimum, which result in PLR–WSVM obtains a better performance.

6. Conclusions

In this paper, PLR–WSVM is used to predict the stock trading signals. Compared with PLR–BNP, PLR–WSVM makes improvements in many aspects. Firstly, each turning point obtained from PLR is set by different weights according to the change rate of the closing price between the current turning point and the next one, which reflects the relative importance of turning point. Secondly, the prediction of trading signals is formulated to a four-class classification problem in which it does not need to determine the threshold of trading decision. Thirdly, WSVM is used to model the prediction of trading signals instead of BPN, which results in the prediction model with a stronger generalization performance. Fourthly, the history data is divided into some overlapping training–testing sets rather than usual training–validation–testing, which not only fully utilizes the data but also reduces the time-variability of the data. In addition, some new technical indicators representing investors' sentiment are added to the input variables, which improve the prediction performance.

The comparative experiments on 20 shares from Shanghai Stock Exchange in China shows the accuracy and profitability of PLR–WSVM are all better than PLR–BPN and BHS, which denotes PLR–WSVM is effective and can provide a higher profitability than PLR–BPN and BHS do.

However, there are still some problems to be studied further in PLR–WSVM.

PLR–WSVM only uses a classifier learned from history data to predict the trading signal. With the time variability of stock data, it may not be the best scheme. Kwon and Moon [13] has shown the effectiveness of ensemble method to stock data. We think the ensemble of multi PLR–WSVM is worth of study in future.

The optimization of the parameters is another problem that needs to be studied further. For example, the profit of PLR–WSVM is impacted by the parameter λ that indicates the relative importance of turning points to the ordinary point. Generally speaking, λ should not be set too big or small. However, how to select the optimal λ is still an open problem.

Finally, the importance and the stability of input variable are also interesting. The importance of one variable means the influence degree to prediction model. It can provide some references to the selection of the optimal subset of input variables. In addition, it is also helpful to the explanation of prediction model. The stability of input variables means the sensitivity of the importance for training data. If the change of the importance of one variable is small when the training data has a small perturbation, then the stability of this variable is good. The stability analysis of input variables is important for stock trading signal prediction. If the stability of one variable is low, then it may not be a good input variable.

Conflict of interest

None declared.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under Grant No. 61271337, the Natural Science Foundations of Fujian Province of China under Grant No. 2011J01373 and the Foundation of Key Laboratory of System Control and Information Processing, Ministry of Education, PR China under Grant No. SCIP2011004.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.asoc.2012.10.026.

References

[1] P.C. Chang, C.Y. Fan, C.H. Liu, Integrating a piecewise linear representation method and a neural network model for stock trading points prediction, IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews 39 (1) (2009) 80–92.
[2] P.C. Chang, T.W. Liao, J.J. Lin, C.Y. Fan, A dynamic threshold decision system for stock trading signal detection, Applied Soft Computing 11 (2011) 3998–4010.
[3] Y.S. Abu-Mostafa, A.F. Atiya, Introduction to financial forecasting, Applied Intelligence 6 (1996) 205–213.
[4] E. Keogh, S. Chu, D. Hart, M. Pazzani, An online algorithm for segmenting time series, in: Proceedings of the IEEE International Conference on Data Mining, 2001, pp. 289–296.
[5] E. Keogh, S. Chu, D. Hart, M. Pazzani, Segmenting time series: a survey and novel approach, in: Data Mining in Time Series Databases, World Scientific Publishing Company, Singapore, 2004.
[6] H. Wu, B. Salzberg, D. Zhang, Online event-driven subsequence matching over financial data streams, in: Proceeding of the SIGMOD, Stream Management, June, 2004, pp. 23–34.
[7] R. Agrawal, C. Faloutsos, A. Swami, Efficient similarity search in sequence databases, in: Proceedings of the 4th Conference on Foundation of Data Organization and Algorithms, 1993.
[8] K. Chan, W. Fu, Efficient time series matching by wavelets, in: Proceedings of the 15th IEEE International Conference on Data Engineering, IEEE Computer Society Press, Los Alamitos, 1999.
[9] R. Agrawal, K.I. Lin, H.S. Sawhney, K. Shim, Fast similarity search in the presence of noise, scaling, and translation in time series databases, in: Proceedings of the 21st International Conference on Very large Data Bases, 1995, pp. 492–500.
[10] G. Das, K. Lin, H. Mannila, G. Renganathan, P. Smyth, Rule discovery from time series, in: Proceedings of the 3rd International Conference on Data Engineering, 1998.
[11] C. Peng, H. Wang, S. Zhang, S. Parker, Landmarks: a new model for similarity based pattern querying in time series databases, in: Proceedings of the 16th International Conference on Data Engineering, 2000.
[12] D. Zhang, L. Zhou, Discovering golden nuggets: data mining in financial application, IEEE Transactions on Systems Man and Cybernetics Part C – Applications and Reviews 34 (November (4)) (2004) 513–522.
[13] Y.K. Kwon, B.R. Moon, A hybrid neurogenetic approach for stock forecasting, IEEE Transactions on Neural Networks 18 (3) (2007) 851–864.
[14] L.J. Cao, F.E.H. Tay, Support vector machine with adaptive parameters in financial time series forecasting, IEEE Transactions on Neural Networks 14 (6) (2003) 1506–1518.
[15] F. Kai, X. Wenhua, Training neural network with genetic algorithms for forecasting the stock price index, in: Proceedings of the IEEE International Conference of Intelligence Processing Systems, 1997, pp. 401–403.
[16] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[17] C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery 2 (2) (1998) 121–167.
[18] B. Schölkopf, A.J. Smola, Learning with Kernels: Support Vector Machines. Regularization Optimization and Beyond, MIT Press, Cambridge, MA, 2002.
[19] C.H. Zhang, Y.J. Tian, N.Y. Deng, The new interpretation of support vector machines on statistical learning theory, Science China Mathematics 53 (1) (2010) 151–164.
[20] L.K. Luo, D.F. Huang, L.J. Ye, Q.F. Zhou, G.F. Shao, H. Peng, Improving the computational efficiency of recursive cluster elimination for gene selection, IEEE/ACM Transactions on Computational Biology and Bioinformatics 8 (1) (2011) 122–129.
[21] C.C. Chang, C.J. Lin, LIBSVM. A Library for Support Vector Machines, 2001, Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm
[22] C.F. Lin, S.D. Wang, Fuzzy support vector machines, IEEE Transactions on Neural Network 13 (2) (2002) 464–471.
[23] J. Weston, C. Watkins, Support vector machines for multi-class pattern recognition, in: Proceedings of the European Symposium on Artificial Neural Networks, vol. 4, no. 6, Citeseer, 1999, pp. 219–224.
[24] P.J. Kaufman, Trading Systems and Methods, Wiley, New York, 1998.
[25] C.W. Hsu, C.C. Chang, C.J. Lin, A Practical Guide to Support Vector ClassificationD 2010 http://www.csie.ntu.edu.tw/∼cjlin
[26] J. Platt, Sequential minimal optimization: a fast algorithm for training support vector machines, in: B. Schölkopf, C.J.C. Burges, A.J. Smola (Eds.), Advance in Kernel Methods – Support Vector Learning, MIT Press, Cambridge, MA, 1999, pp. 185–208.
[27] Keerthi, et al., Improvements to Platt's SMO algorithm for SVM classifier design, Neural Computation 13 (2001) 637–649.
[28] R.E. Fan, P.H. Chen, C.J. Lin, Working set selection using second order information, Journal of Machine Learning Research 6 (2005) 1889–1918.
[29] S. Alexander, Price movements in speculative markets: trends or random walks, Industrial Management Review 2 (1961) 7–26.
[30] E.F. Fama, M.E. Blume, Filter rules and stock market trading, Journal of Business 39 (1966) 226–241.
[31] F. Allen, R. Karjalainen, Using genetic algorithms to find technical trading rules, Journal of Financial Economics 51 (1999) 245–271.
[32] M.J. Ready, Profits from technical trading rules, Financial Management 31 (2002) 43–61.