A Framework for Optimization of Pattern Sets for Financial Time Series Prediction

Mattias Wahde(✉)

Chalmers University of Technology, 412 96 Göteborg, Sweden
mattias.wahde@chalmers.se

Abstract. In this paper, a framework is introduced for generating human-interpretable structures, here called *pattern sets*, for short-term prediction of financial time series. The optimization is carried out using an evolutionary algorithm, which is able to modify both the structure and the parameters of the evolving pattern sets. The framework has been applied in two different modes: A tuning mode, in which the user provides a starting point in the form of loosely defined pattern set, and a discovery mode, in which the starting points consist of random pattern sets. The best results were obtained in the tuning mode, for which the top-performing pattern sets gave strongly statistically significant results in excess of one-day market returns (p-values below 0.0007 and, in many cases, even below 0.0001) over validation data (not used during optimization) for two data sets, involving stocks with large and small market capitalization, respectively. The average one-day returns ranged from 0.518 to 1.147%, with one-day Sharpe ratios ranging from 0.138 to 0.258.

Keywords: Financial prediction · Time series analysis · Evolutionary computation

1 Introduction

On a typical day, millions of financial instruments change hands in the world's financial markets. In many cases, though by no means all, the decision to buy or sell a financial instrument is based primarily on an analysis of the price time series for the instrument in question, i.e. the sequence of preceding prices over some time interval. In a broad sense, analysis and prediction of financial time series occurs on all levels, from the formal analysis carried out in academic research to the rapid, and much less formal, financial predictions continuously available on TV and on the internet.

However, whereas many traders, fund managers, analysts, and other financial professionals typically base their predictions and actions on what is known as *technical analysis*, i.e. the interpretation of stock price patterns, academic researchers are often strongly sceptical to such approaches. This scepticism has both a theoretical and an empirical foundation. On the theoretical side, researchers refer to the *efficient market hypothesis* (EMH) (see e.g. [1,2]), which

Y. Bi et al. (eds.), *Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016*,
Lecture Notes in Networks and Systems 15, DOI 10.1007/978-3-319-56994-9_2

essentially says that all available information is reflected in the price of a financial instrument, so that nothing can be gained by analysing the price patterns as a basis for prediction. On the other hand, the EMH has been challenged in view of the sometimes irrational and emotional behavior of traders [3], as studied in the field of behavioral finance.

On the empirical side, using time series analysis, a large variety of technical trading rules have been thoroughly tested by many authors. For reviews see, for example, [3,4]. In careful analyses, technical indicators such as, for example, those based on (versions of) moving-average crossings, are generally found to have very little predictive value, even though some predictability has been found in the context of certain specific markets, as discussed at length in the reviews just mentioned.

As an alternative to considering standard time series indicators of the kind just mentioned, financial traders often make use of a so-called candlestick representation, briefly described in Subsect. 2.1 below, where the price data over a given period (for example, one day) is compressed into four numbers (open, high, low, close) that, moreover, can be given a visual representation, also briefly described below. Interpreting the patterns that appear when a time series is represented in this way is a common approach in financial trading. Alas, such patterns, whether bullish (i.e. predicting a higher future price) or bearish (predicting a lower future price) are no better at predicting prices, at least when considered separately. However, simple patterns would not, by themselves and in isolation, normally be used by financial traders. Instead, traders focus on a combination of factors such as, for example, a supposedly bullish price pattern occurring at the end of an often rather vaguely defined downtrend in prices.

For cases involving candlestick patterns indicative of a trend reversal, using daily stock data Caginalp and Laurent [5] found a statistically significant (two-day) predictability, albeit based on rather short time series. Many other studies on candlestick patterns have been presented over the years, some finding predictive power in certain patterns (see e.g. [6]), and others finding no such results (see e.g. [7]). Despite the disparate findings, these studies are not necessarily contradictory, since they generally concern different data sets and make use of diverse approaches: In some studies, pre-defined, handcrafted patterns have been applied to a given data set in order to test for statistically significant predictive power, whereas in other studies (typically the ones involving optimization) a set of trading rules has been optimized on one data set and then tested, as it should be, on another data set. Moreover, different authors have used different holding periods when applying stock patterns in trading and also somewhat dissimilar definitions of the price patterns, e.g. the exact definition of a price *trend*. In summary, it is probably fair to conclude that the predictive power of candlestick patterns is unclear.

Since the performance of price patterns certainly may depend on their exact definition, a valid approach to finding predictability in financial time series is to define a set of generic rules, for example a set of parameterized technical trading rules, and then apply some form of optimization to select a suitable set of rules and also to set their parameters. Such studies, of which many have been conducted, often make use of stochastic optimization methods, for example based on the general framework of evolutionary computation and, in particular, genetic algorithms (GAs) [8] and genetic programming (GP) [9]. In their standard form, GAs are used for carrying out parametric optimization, by artificial evolution of strings of digits representing the parameters of a given structure. GP, by contrast, carries out parametric *and* structural optimization, by searching for tree-like structures involving combinations of various operators such as standard arithmetic operators and, in the case of financial prediction, also time-series related operators for generating, say, moving averages, as well as numerical parameters used in the structures. A recent and thorough review of evolutionary approaches in financial trading is available in [10].

Genetic programming (GP) provides an open-ended approach to optimization, in which the complexity of the generated structure has no (explicit) bound and the resulting trading rules can therefore often be beyond human interpretability. This, in itself, is not necessarily a problem, but it does raise the question whether the trading rules found in this manner really provide a test of the predictive power of the price patterns applied by human traders [3]. By contrast, the main aim of this paper is to introduce and describe a framework for open-ended structural and parametric optimization of *human-interpretable* structures for financial prediction (henceforth referred to as *pattern sets*). Moreover, with this framework, it will be possible both to extend (structurally) and to modify (parametrically) a given pattern set, while maintaining a high degree of interpretability. Thus, for example, the user may seed the optimizer with a pattern set loosely defining, say, a falling trend followed by some bullish candlestick pattern, and then let the optimizer modify the pattern set structurally and parametrically, removing human bias that might have been introduced in the original pattern set. Alternatively, one may let the optimizer generate its own initial random population of pattern sets which are then optimized without any restriction on the structure. A secondary aim of the paper is to analyse the performance of such pattern sets (particularly the first kind, where the user provides a starting point) in predicting one-day returns on daily stock data from the US markets.

The outline of the paper is as follows: Sect. 2 gives a brief introduction to financial time series and candlestick patterns and also describes the data used in this study. The method is outlined in Sect. 3. Next, in Sect. 4, the results are presented, followed by a discussion in Sect. 5. The conclusions are presented in Sect. 6.

2 Data

2.1 Candlestick Representation of Financial Time Series

The standard way of representing a (one-dimensional) time series is, of course, as a list of time-value pairs. However, in finance, it is common instead to use a so-called candlestick representation, as illustrated in Fig. 1. In this representation,

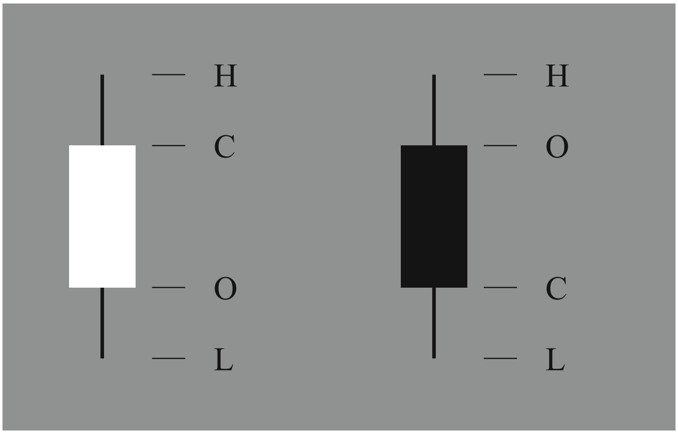

Fig. 1. The definition of candlesticks. The open price is denoted O and the close price (at the end of the period over which the candlestick is defined) is denoted C. If the close price is larger than the open price, the candlestick is drawn with a white body (left part of the figure), otherwise it is drawn in black (right). The highest price is denoted H and the lowest price is denoted L. The vertical lines emanating from the candlestick bodies are called the upper shadow and the lower shadow, respectively. In the figure, the upper and lower shadows have equal length, for clarity, but this is of course not generally the case.

which traces its origins to the rice trade in Japan in the 18$^{\text{th}}$ century, for a given time interval (hereafter enumerated with the index i), the price information is compressed to only four numbers: The *open* price O_i at the beginning of the interval, the maximum (*high*) price H_i, the minimum (*low*) price L_i, and the *close* price C_i at the end of the interval. The *body* of the candlestick, hereafter called B_i, is defined as $C_i - O_i$. Note that B_i can thus be either positive or negative (or zero). The *body top*, here denoted B_i^{t} is equal to $\max(\mathrm{O_i}, \mathrm{C_i})$, whereas the *body bottom*, here denoted B_i^{b} equals $\min(\mathrm{O_i}, \mathrm{C_i})$. Also, the *upper shadow* of a candlestick is defined as $S_i^{\mathrm{U}} = H_i - B_i^{\mathrm{t}}$, and the *lower shadow* is defined as $S_i^{\mathrm{L}} = B_i^{\mathrm{b}} - L_i$. Two additional definitions are needed for the investigation presented here: For a given candlestick, dropping the index i for clarity, the six quantities $\{O, H, L, C, B^t, B^b\}$ are henceforth collectively referred to as *candlestick values*, a set denoted $\mathcal{C}_{\mathrm{v}}$, whereas the three quantities $\{B, S^U, S^L\}$ are collectively referred to as *candlestick parts*, a set denoted $\mathcal{C}_{\mathrm{p}}$.

2.2 Data Selection

Two different data sets were generated, using daily (rather than intraday) stock market data. The first set, $\mathcal{D}_1$ contained price data for 60 large US companies, in this case with a market capitalization (in early February, 2016) of around 50 billion USD or more, whereas the second set $\mathcal{D}_2$ consisted of price data for 60 (relatively) small companies, with a market capitalization between 1.7 billion USD and

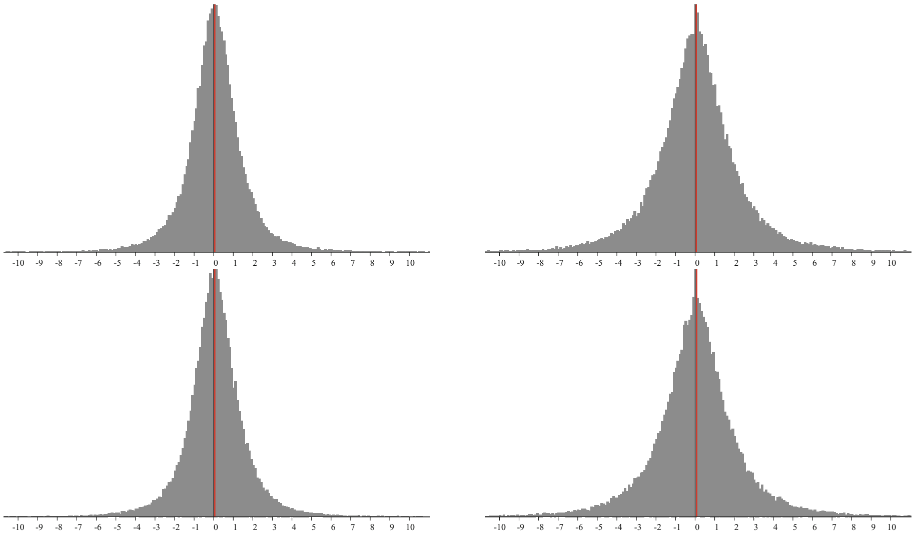

Fig. 2. The distribution of one-day returns (measured in per cent) for each data subset. The panels on the left correspond to the first data set ($\mathcal{D}_1$), with the training set ($\mathcal{T}_1$) shown in the upper panel and the validation set ($\mathcal{V}_1$) shown in the lower panel. The panels on the right show the distributions for the second data set ($\mathcal{D}_2$), again with the training set ($\mathcal{T}_2$) in the upper panel and the validation set ($\mathcal{V}_2$) in the lower panel. The red lines indicate the averages of the distributions, the values of which are given in Table 1. The bin width of the histograms was set to 0.1%. The vertical scale is somewhat arbitrary, but if the values are normalized by the total number of samples (83640), the distributions can be seen as probability distributions.

2.0 billion USD. For each stock, the corresponding time series contained data from January 2005 until February 2016, 2789 days in total. All time series were adjusted for dividends and splits. The two data sets were then each further divided into two subsets, a training set $\mathcal{T}_i$, $i = 1, 2$, and a validation set $\mathcal{V}_i$, $i = 1, 2$, each containing data from 30 stocks, without any overlap between the two subsets. Thus, $\mathcal{D}_i = \mathcal{T}_i \cup \mathcal{V}_i$, where $\mathcal{T}_i \cap \mathcal{V}_i = \emptyset$, $i = 1, 2$. Each of the four subsets thus generated contained a total of 83670 data points. For every data subset, the one-day returns, defined as

$$r_{1,i} = \frac{C_{i+1}}{C_i} - 1, \;\; i = 1, 2, \ldots \tag{1}$$

were computed. With 30 data series and 2789 data points in each, one thus obtains $30 \times 2788 = 83640$ one-day returns. These distributions, measured in per cent (i.e. $r_{1,i} \times 100$) are shown as histograms in Fig. 2. The average return for each data subset is given in Table 1. As can be seen in the table, the averages are slightly above 0, indicating that, over the 11 years spanned by the time series, the market prices have generally risen, albeit with many fluctuations along the way.

Table 1. The average and standard deviation of the one-day returns for each data subset.

Data subset	$\overline{r}_1$ (%)	st. dev. (%)
$\mathcal{T}_1$	0.0515	1.95
$\mathcal{V}_1$	0.0538	1.96
$\mathcal{T}_2$	0.0544	2.72
$\mathcal{V}_2$	0.0687	3.01

3 Method

For any given situation, the pattern sets introduced here either suggest an action or they do not, i.e. they make a binary decision. The action consists of entering into a *long position*[1] at the close of the current candlestick (relative index 0), and exiting at the close of the next candlestick (relative index 1). In other words, the pattern sets make a one-step prediction that the price will rise. Of course, the pattern sets can easily be applied to longer predictions, but here all predictions will involve a single step. In the representation used here, a pattern set consists of a sequence of (candlestick) *patterns*, each of which is defined as a ratio between two quantities, as described below, along with an interval $[a, b]$ in which the ratio should fall for the pattern to match. The pattern set as a whole will suggest an action only if *all* its constituent patterns match.

3.1 Pattern Types

Three different candlestick pattern types have been defined, namely (i) the *Part pattern* (PP), (ii) the *Value pair pattern* (VPP), and (iii) the *Body pair pattern* (BPP).

The PP involves a single candlestick, with a given index *relative* to the current location along the time series, where index 0 denotes the current candlestick and where negative indices are used for preceding candlesticks. This pattern type compares two parts (see the definition in Subsect. 2.1) of the same candlestick: Two candlestick parts, $P1$ and $P2$, are thus selected from the set $\mathcal{C}_{\mathrm{p}}$ for the candlestick in question, and the ratio $P1/P2$ of their values is computed. If and only if this ratio falls within a given range $[a_{\mathrm{p}}, b_{\mathrm{p}}]$, for a given location along the time series, the pattern is said to *match*. As a specific example, if $P1 = S_0^U$, $P2 = B_0$ (thus representing the upper shadow and the body of the current candlestick), and the range is set to $[1, 2]$, this pattern would require that the upper shadow of the candlestick should be at least as large as the body of the candlestick, but no more than twice as large.

[1] In a long position, a profit is made if the price rises. Short positions, where a profit is made if the price falls, will not be considered here, but could certainly be included in principle.

The BPP and the VPP, by contrast, involve two candlesticks typically, but not necessarily, with different indices. The VPP compares two candlestick *values* (again, see the definition in Subsect. 2.1) $V1$ and $V2$. In this case, the ratio $V1/V2$ is computed, and it is then compared with a range $[a_{\mathrm{v}}, b_{\mathrm{v}}]$. Again, if the ratio falls within this range, the pattern matches, otherwise it does not. As a specific example, consider a pattern which compares the body bottom (B_0^{b}) of the current candlestick to the body top (B_{-1}^{t}) of the previous candlestick, with a range of $[1.01, 1.10]$. In this case, the pattern matches if there is a positive gap of at least one per cent, and at most ten per cent, between the body of the preceding candlestick and that of the current candlestick.

Finally, the BPP compares the body of one candlestick to that of another candlestick. Once again, if that ratio falls within a given range $[a_{\mathrm{b}}, b_{\mathrm{b}}]$ the pattern matches. Thus, here, the ratio $B1/B2$ is computed, using the body sizes (which can be negative) of two candlesticks. For example if B1 is taken as the body of the current candlestick and B2 as the body of the preceding candlestick, a range of $[0.5, 1]$ would require the body size of the current candlestick to be at least half as large as, and at most equal to, the body size of the preceding candlestick.

In general, for all pattern types, if the quantity in the denominator (of the ratio) is equal to zero, the pattern is considered *not* to match. Note also that a pattern involving the ratio between any two quantities can be replaced by a pattern involving the inverse ratio (and, of course, then the inverse range limits), so that one can always define a pattern requiring any particular quantity being equal to zero.

Combining sequences of patterns selected from the three types above, one can represent a very wide range of time series situations. As one example, among many, see the loose definition of a downtrend described in Subsect. 4.1 below. Importantly, in addition to a high degree of versatility, this representation also has a high degree of *interpretability*.

3.2 Encoding

In order to use the patterns in optimization, one must also have a procedure for encoding all possible patterns. This procedure, which will be described next, is fairly straightforward, but the description below might appear somewhat daunting. Thus, a specific example that the reader may wish to consult while reading is given at the very end of the subsection.

The three patterns described above can each be encoded as a set of numbers. Starting with the most complex pattern, the VPP, seven numbers are necessary: An integer defining the type of the pattern (in this case, VPP) as described in the caption of Table 2, two integers determining the indices of the two candlesticks, such that a value of 0 indicates the current index and negative values indicate earlier indices, two integers determining which candlestick values to use, from the set $\mathcal{C}_{\mathrm{v}}$ (see Subsect. 2.1 above), and two floating-point numbers determining the range $[a_{\mathrm{v}}, b_{\mathrm{v}}]$. For the PP, only six parameters are needed: One integer defining the type of the pattern, another integer defining the candlestick index relative to the current index, two integers determining which candlestick parts to use, from the set $\mathcal{C}_\mathrm{p}$, and two floating-point numbers determining the range $[a_\mathrm{p}, b_\mathrm{p}]$. Finally, for the BPP, only five parameters are needed: Again, one integer defining the pattern type, two integers determining the indices of the two candlesticks and two floating-point numbers defining the range $[a_\mathrm{b}, b_\mathrm{b}]$.

Now, when evolving pattern sequences as described below, a crucial feature of the optimizer will be its ability not only to change parameters of existing patterns, but also to transform a given pattern (a PP, say) to a pattern of another kind (VPP, for example). Thus, a uniform encoding is required, such that every pattern encoding has the same length and every number in the encoding has the same range. If this is the case, it is possible to change the first number in the encoding of a pattern, namely the one defining the type of pattern, and still get a valid pattern, but of a different type. For that reason, the patterns are all encoded using seven numbers: Five integers and two floating-point numbers. For the VPP all numbers are used. For the PP, the third number, which for the VPP defines the second candlestick index, is just treated as an unused dummy parameter, since the PP requires only one candlestick index. Similarly, for the BPP, the fourth and fifth numbers are unused dummy parameters since, for this pattern, it is known *a priori* that the respective candlestick bodies are to be used.

The pattern encoding scheme is summarized in Table 2. It should be noted that, in order to make it possible to change the type of the pattern, the range of any parameter must enclose the maximum possible range of that parameter for any pattern. Thus, for example, the fourth and fifth parameters, which define the candlestick values for the VPP and the candlestick parts for the PP take the range $[0, 5]$ (since there are 6 candlestick values but only 3 candlestick parts, as explained in Subsect. 2.1 above). Thus, when used in a VPP, the number is simply read off and the corresponding candlestick value is chosen: If the number is 0, the open (O) is chosen, if it is 1, the high (H) is chosen etc. By contrast, in

Table 2. Encoding scheme for patterns. For the pattern type (the first parameter), a PP is generated if the value is 0, a VPP if the value is 1, and a BPP if the value is 2. The fourth and fifth parameters determine candlestick values in the case of a VPP and candlestick parts for a PP. In the latter case, the numbers are taken modulo 3, to get a value in the range $[0, 2]$. See the main text for a complete description of the encoding.

Parameter	Usage	Range
1	Pattern type definition	$[0, 2]$
2	First index definition	$[-20, 0]$
3	Second index definition	$[-20, 0]$
4	First component identifier	$[0, 5]$
5	Second component identifier	$[0, 5]$
6	Lower range limit (a)	$[-10, 10]$
7	Range width ($b - a$)	$[0, 20]$

a PP, the number is taken *modulo 3*, thus giving a value in the range $[0, 2]$ that, in turn, can be used for identifying the candlestick part. Moreover, the ranges $[a, b]$ are defined by specifying a minimum value, corresponding to a, and the width of the interval, i.e. $b-a$, rather than specifying both a and b directly. Note also that a can be either positive or negative, even though negative values are only relevant in the case of the PP and BPP types. Negative values are simply ignored for the VPP type, thus making the effective lower range limit equal to 0 for that particular pattern type.

As a specific example, consider the parameter sequence $1, -2, -5, 3, 1, 1.0, 0.2$. This sequence would be decoded as follows: The first parameter (1) identifies the pattern as a VPP; see also Table 2. The second and third parameters identify the candlestick indices (-2 and -5) relative to the current index. The fourth parameter determines which candlestick value to use for the first candlestick in the pattern. In this example, the number is 3, so that the fourth element in the set $\mathcal{C}_{\mathrm{v}}$ should be used, i.e. the close C of the corresponding candlestick. Similarly, the fifth parameter in the sequence (1) identifies the candlestick value for the second candlestick as the second element in $\mathcal{C}_{\mathrm{v}}$, namely the high value H. The two last parameters determine the allowed range as $[1.0, 1.2]$, noting that the last parameter defines the width of the interval $(b - a)$. Thus, to summarize, this pattern would compute the ratio C_{-2}/H_{-5}, i.e. the ratio between the close value two steps before the current index and the high value five steps before the current index. If, for a given current index, the ratio falls within the range $[1.0, 1.2]$, the pattern would match, otherwise it would not.

3.3 Evaluation

The evaluation of a pattern set is very straightforward: For every index in each of the time series in the data subset in question, the evaluation function runs through the pattern set, checking each pattern for a match. Note that, in the first steps of a time series, some patterns cannot be computed. For instance, the pattern defined in the example at the end of the previous section can only be computed from index 5 in the time series, and onwards, since it requires the value H_{-5} etc.

For any index i such that all the patterns in the set provide a match (referred to as a *matching index* for the pattern set in question), the one-day return $r_{1,i}$ is computed and added to a list. Thus, at the end of a pattern set evaluation, the evaluation function will contain a distribution of one-day returns. The objective function used in optimization (see below) is taken as the Sharpe ratio (S) of the distribution of one-day returns, defined as

$$S = \frac{\overline{r}_1 - \rho_1}{\sigma}, \tag{2}$$

where $\overline{r}_1$ is the average over the distribution of one-day returns, ρ_1 is the return of a risk-free asset, typically taken as the 3-month US Treasury bill rate, and σ is the standard deviation of the distribution of one-day returns. This commonly used metric thus computes the excess return over a risk-free asset, in relation to its variability. The risk-free return is currently near 0, but has of course varied over the years. Here, a value of 2% per annum, corresponding to 0.00786% per (trading) day, was chosen.

3.4 Evolutionary Algorithm

The optimization of pattern sets has been carried out using an evolutionary algorithm (EA), implemented in C# .NET. As in any EA, this algorithm maintains a population of M (here usually 100) individuals, in this case each defining a pattern set. The individuals are encoded in chromosomes (strings of digits) consisting of sequences of length $K = 7N$, where N is the number of patterns in a given pattern set, remembering that each pattern is defined using 7 numbers. For a given individual, the decoding procedure thus results in a set of N patterns, which is then evaluated over the data subset in question, resulting in a single number, namely the Sharpe ratio, which is taken as the fitness value. However, for pattern sets that yield only a small number of matching indices, a multiplicative penalty term is applied. This term is equal to 1 for an evaluation resulting in a pre-specified minimum number of matching indices (here taken as 500) or above, and smaller than 1 otherwise.

Selection is carried out using standard tournament selection, with a given tournament size S_{t} (typically 5–10) and a tournament selection parameter p_{t} (around 0.7–0.8). Crossover is applied with probability p_{c} (around 0.7–0.9) to a pair of selected individuals. Two different crossover procedures have been defined, namely a length-preserving crossover applied to chromosomes of equal length, and a non-length-preserving crossover resulting in chromosomes of unequal length. The use of each crossover method is described in Sect. 4 below, and the two methods are also illustrated in Fig. 3. Note that crossover points are always (randomly) selected between pattern encodings, i.e. at positions 7, 14, 21 etc. along a chromosome. Once two new, offspring individuals have been formed, they are subjected to

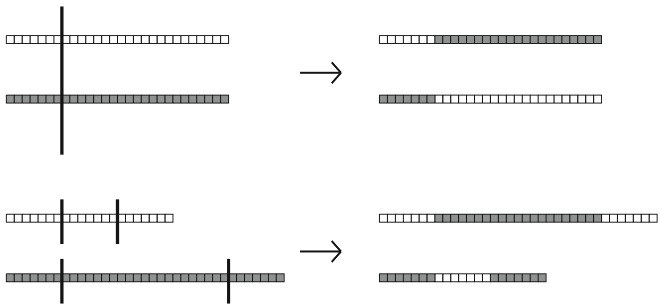

Fig. 3. The two crossover methods used here. Top panel: Length-preserving crossover, using a single crossover point applied to chromosomes of equal length. Bottom panel: Non-length-preserving two-point crossover that generally, but not always, results in chromosomes of different length.

random mutation, using a mutation rate of p_{rel}/K, where p_{rel}, the relative mutation rate, is normally around 1 so that, on average, one mutation occurs per chromosome.

4 Results

The method described above has been applied in two different ways, namely (i) a *tuning mode*, in which the user provides a rather vaguely defined starting point for the optimization and (ii) a *discovery mode* in which the starting point is random and the size of the evolved pattern sets has no upper limit.

4.1 Tuning an Existing Pattern Set

The tuning mode is suitable in cases where one wants to investigate whether a particular situation, such as a downtrend interrupted by a supposedly bullish pattern, yields positive results above market returns. In this mode, the user specifies a set of k patterns defining some particular situation. Then another $N - k$ patterns are added such that, at the starting point of optimization, those added patterns trivially match. Such patterns can easily be defined: For example, a VPP in which the values are taken as $V1 = V2 = C_0$, so that the computed ratio is equal to $C_0/C_0 \equiv 1$, will always match if the range is set, for example, to $[1 - \epsilon, 1 + \epsilon]$, for any $\epsilon \geq 0$. Thus, since the added patterns trivially match, they will not influence the computation while, at the same time, giving the optimizer the *possibility* of making use of those patterns later on, if needed. This is so since, through mutation, an initially trivially matching pattern can be modified into a pattern that only matches a very specific set of situations. Moreover, in the tuning mode, length-preserving crossover (defined in Subsect. 3.4) is used since, otherwise, the initial patterns provided by the user would quickly be destroyed through crossover.

As a specific application example, consider a situation in which a user believes that a short-term (a few days) downtrend, interrupted by some as yet unspecified pattern, will predict a subsequent price increase. In this case, the user can set up a pattern set with a few patterns loosely defining a downtrend, then add a few trivially matching patterns as just described, and start the optimizer, which will then be able to both fine-tune the trend definition *and* find a set of suitable bullish patterns (possibly a single pattern) marking the end of the trend. Here, as an example, a five-day trend was defined using three VPPs, computing the ratios C_{-3}/C_{-1}, C_{-4}/C_{-2}, and C_{-5}/C_{-3}, respectively, and giving a match if those ratios all fell in the range $[1.003, 2]$. As shown in Fig. 4, those three patterns generally define a short-term downtrend, even though one can certainly find situations in which all patterns match even if the downtrend is barely discernible, as can be seen in the lower right panel of the figure. Moreover, the trend may of course extend further back in time, beyond relative index -5, since the three defined patterns do not involve such indices. Next, a few (typically 2–6) trivially

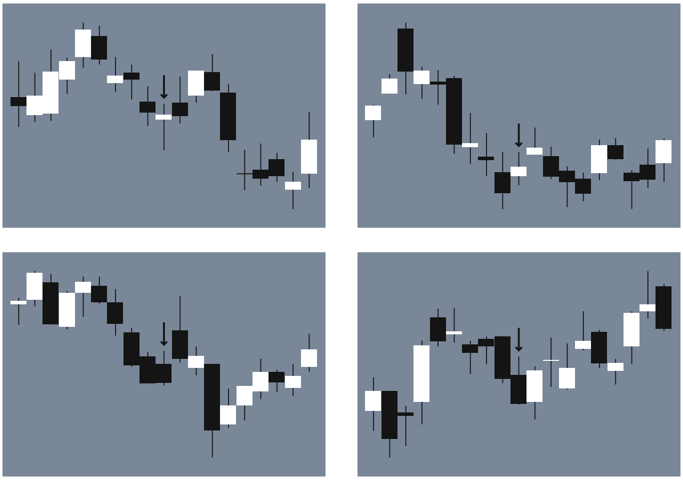

Fig. 4. A few *examples* of five-day downtrends found in the data subset $\mathcal{T}_1$. The trends were defined using three patterns, as described in the main text. The arrows indicate the current index (relative index 0). The trends cover relative indices -5 to -1. Note that, initially, the pattern ending the trend (at relative index 0) is unspecified, and may thus take any form.

matching patterns were added, and M identical copies of the resulting pattern set were made, forming the initial population.

Here, the user thus only provided a *starting point* for the optimizer: The fact that the downtrend is initially loosely defined is precisely the point, since it is the task of the optimization method to determine a detailed definition of the downtrend and also to find suitable pattern(s) that should mark the end of the trend, in order for the resulting pattern set to generate a profitable one-step prediction. Note that the optimizer can also use the added patterns to extend the trend. Thus, for example, one of the added initially matching patterns could be modified into a VPP defining the ratio C_{-6}/C_{-4}, thus (with proper range limits) extending the trend one more step backwards. The optimizer can of course also tune the definition of the trend by changing, for example, the range limits or indeed the values used. For example, it may be found that L_{-3}/C_{-1} gives better results than C_{-3}/C_{-1} etc.

Several optimization runs were carried out, using slightly different EA parameters and different random number generator seed values. In order to generate some initial diversity, all but the first individual were also subjected to a mutation step (with probability 1) before starting the actual optimization. A summary of the results obtained for the best pattern set in five different runs is given in Table 3. In all cases, only the training sets ($\mathcal{T}_1$ or $\mathcal{T}_2$) were used during optimization. The best pattern set found was then also applied to the corresponding validation set ($\mathcal{V}_1$ or $\mathcal{V}_2$). In order to determine whether the distribution of results obtained over previously unseen data represented a statistically significant improvement over the

Table 3. Results obtained for the tuning runs, starting from a downtrend defined by the user as described in the main text. The first two columns indicate the pattern set ID and the data subsets used (training and validation). The next three columns show the number of instances found (#), the average one-day return ($\bar{r}_1$), and the Sharpe ratio (S) for the *training set*. The next three columns show the same quantities for the *validation set*. The final two columns give the results of the Wilcoxon test described in the main text.

		Training			Validation				
ID	Tr., Val	#	$\bar{r}_1$	S	#	$\bar{r}_1$	S	z	p
PS1	$\mathcal{T}_1$, $\mathcal{V}_1$	641	0.758	0.243	609	0.775	0.258	6.93	<0.0001
PS2	$\mathcal{T}_1$, $\mathcal{V}_1$	518	0.626	0.201	456	0.518	0.174	3.74	<0.0001
PS3	$\mathcal{T}_1$, $\mathcal{V}_1$	491	0.667	0.207	484	0.781	0.201	6.86	<0.0001
PS4	$\mathcal{T}_2$, $\mathcal{V}_2$	657	1.038	0.187	599	1.147	0.184	4.21	<0.0001
PS5	$\mathcal{T}_2$, $\mathcal{V}_2$	602	0.916	0.186	547	0.550	0.138	3.21	<0.0007

one-day returns obtained by the market as a whole, a standard Wilcoxon rank-sum test was carried out, comparing the distribution of one-day validation results for the evolved pattern set to the distribution of all one-day returns for the validation set in question (see also the two lower panels in Fig. 2). The rightmost two columns in Table 3 give the corresponding z-scores and p-values, the null hypothesis being that the two distributions are equal, and the alternative hypothesis being that the distribution for the evolved pattern set is shifted to the right (positive values) relative to the distribution of all one-day returns.

As an example, Fig. 5 shows the distribution obtained for PS1 from Table 3 over the validation set $\mathcal{V}_1$. In this particular case, a comparison with the market return for the validation set, i.e. the distribution shown in the lower left panel of Fig. 2, indicated that the pattern set gives a strongly significant positive result, with an average one-day return of 0.775% and a p-value of 2.1×10^{-12}. Even though it might not be immediately evident from the rather noisy plot shown in Fig. 5, the distribution is, in fact, rather strongly skewed towards positive values, with a total of 11 instances giving one-day returns above 10% and the best instance having a one-day return of 23.4%. The pattern set (PS1) still used three patterns to define the downtrend. However the trend was extended to cover relative indices -6 to -1. Moreover, the limits were somewhat modified, as were the candlestick values used to define the trend. For example, one of the constituent patterns compares L_{-4} to C_{-2} rather than comparing C_{-4} to C_{-2} as in the initial pattern set. Interestingly, the optimizer also defined a specific pattern to mark the end of the trend, requiring that $S_0^{\mathrm{U}}/S_0^{\mathrm{L}}$ should fall in the range $[5.01, 24.1]$. This pattern is reminiscent of what a trader would refer to as an *inverted hammer*. A few of the 609 instances (for the validation set) are shown in Fig. 6.

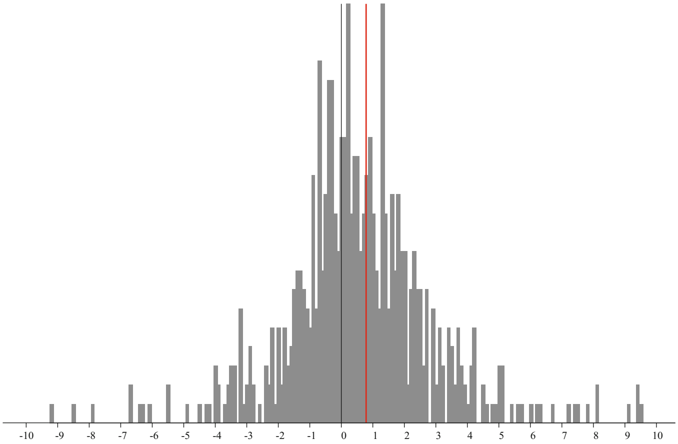

Fig. 5. The distribution of one-day returns obtained over the validation set for PS1. The average one-day return, indicated by a red vertical line, is equal to 0.775%. Note that the 11 most positive instances (returns above 10%) are not shown in the figure. The most negative instance (with a return of −9.16%) does appear in the figure, however.

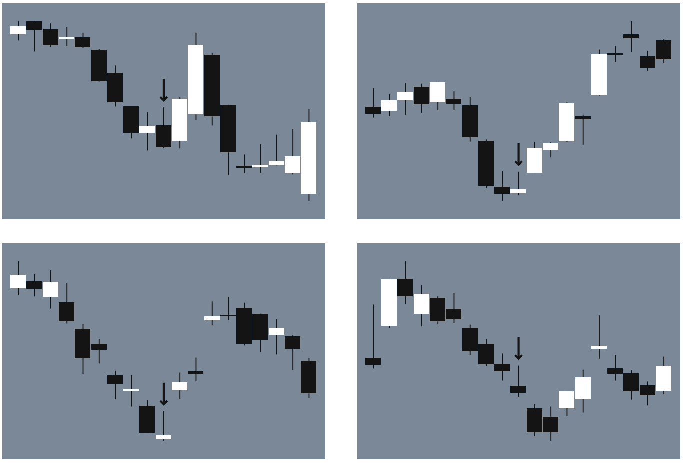

Fig. 6. A few *examples* of the instances found by the pattern set PS1; see also Table 3 and the main text. Three positive instances are shown, and one negative instance (lower right panel). As can be seen, this pattern set requires that, in addition to the downtrend (whose definition also was modified by the optimizer), a candlestick with a strong upper shadow should occur for relative index 0, indicated by the vertical arrow.

As can be seen in Table 3, similar results were obtained for the other runs but, of course, with different pattern sets. For example, PS2 extended the trend to cover indices −14 to −1. Regarding the pattern sets evolved over the second training set ($\mathcal{T}_2$), PS4, for example, used a slightly modified five-day trend, but also added two conditions that, together, required the current candlestick (relative index 0) to have a negative body with a large upper shadow. For all the presented pattern sets, the null hypothesis can be strongly rejected, with $p-$values smaller than 0.0007, meaning that the patterns sets give one-day returns in excess of market returns, over their respective validation sets. In general, the pattern sets evolved using $\mathcal{T}_2$, i.e. PS4 and PS5, give higher average returns over the training set than the patterns set PS1–PS3 evolved with $\mathcal{T}_1$. However, PS4 and PS5 also have lower Sharpe ratios over the training set. PS4 has a rather high Sharpe ratio over the validation set, albeit lower than those of PS1 and PS3.

4.2 Discovering New Pattern Sets

In the discovery mode, the starting point is a population of randomly generated pattern sets (with an initial maximum number of constituent patterns, here equal to 10). During optimization, two-point crossover is used, meaning that there is no restriction on the structure of the generated pattern sets. Also in this case, several runs were carried out, and some of the results are summarized briefly in Table 4. In these runs, the results were generally less good than for the tuning runs presented above: Both the average one-day returns and the Sharpe ratios were lower, especially over the validation set; see also the discussion below. In fact, for PS7, the results over the validation set are just barely statistically significant. A detailed description of the pattern sets will not be given here. Suffice it to say that the structures obtained were generally more complex than for the tuning runs, but still interpretable.

Table 4. Results obtained for the discovery runs. The columns are the same as in Table 3.

		Training			Validation				
ID	Tr., Val	#	$\overline{r}_1$	S	#	$\overline{r}_1$	S	z	p
PS6	$\mathcal{T}_1$, $\mathcal{V}_1$	531	0.309	0.182	590	0.199	0.135	2.75	<0.003
PS7	$\mathcal{T}_2$, $\mathcal{V}_2$	512	0.424	0.173	483	0.250	0.087	1.65	0.0499

5 Discussion

The results presented above indicate that the framework introduced here is capable of generating pattern sets giving statistically significant results above (one-day) market returns over previously unseen validation data. The best-performing pattern set, based on Sharpe ratios, was PS1. Its daily Sharpe ratio over the validation set of around 0.258 corresponds to a respectable annualized Sharpe ratio of $0.258\sqrt{252} = 4.1$ that, for example, is at the high end of the performance of trading rules reviewed by Bajgrowicz and Scaillet [4]. However, one should be careful not to directly extrapolate the results obtained here to an actual trading situation, partly because trade execution (buying very near the close of the current day) and trading costs have not been considered here, and partly because obtaining such an annualized Sharpe ratio would require trades occurring every day, which might not be the case, given the number of instances found.

For both data sets, the results obtained over the validation sets were in excess of market returns, with a high degree of statistical significance. One may, however, be concerned that the tests of statistical significance considered above are in fact likely to give such results, since the stocks in the two pairs of training and validation sets ($\mathcal{T}_1$ vs. $\mathcal{V}_1$ and $\mathcal{T}_2$ vs. $\mathcal{V}_2$) do behave rather similarly, especially in the case of larger companies. Thus, as an additional test, the PS1 pattern set was applied to the *other* validation set, i.e. $\mathcal{V}_2$. Somewhat surprisingly, even in this case, a strongly positive result was obtained, namely an average one-day return ($\bar{r}_1$) of 0.647% and a Sharpe ratio of 0.152, and with a p-value for the Wilcoxon test smaller than 0.0001. Similarly, the PS4 pattern set was applied to $\mathcal{V}_1$. Also in this case, excellent results were obtained: $\bar{r}_1$ was equal to 1.73%, with a Sharpe ratio as high as 0.307, and a p-value smaller than 0.0001. However, for PS4, the number of instances was rather small (329). Still, these two tests indicate that the best pattern sets obtained in the tuning mode, PS1 and PS4, are indeed able to capture important aspects of the time series for making one-day predictions.

The results presented here were obtained in runs lasting up to a few hours. An interesting avenue for further work would be to increase the size of the data sets by one or two powers of ten, by adding more stocks and by extending the length of the time series. With such large data sets a faster optimization process would be needed. Therefore, research is underway to generate a parallel, asynchronous, client-server version of the EA, with the aim of speeding up the optimization process.

A crucial aspect of the optimization framework or, rather, the encoding procedure, is the fact that the optimizer is allowed to modify both the parameters *and* the structure of the evolving pattern sets. This applies to both running modes: Even in the case of the tuning mode, where the resulting pattern sets kept aspects of the original, user-defined pattern set they also contained new aspects, generated during optimization by changing the identity of some patterns (e.g. from a PP to a VPP). The fact that the tuning mode generally gave better results than the discovery mode does not imply that the latter should not be used. Instead, the difference follows as a rather natural consequence of the very much larger size of the search space when running in discovery mode. In fact, the range of possible structures for the tuning mode is a subset of the complete range of structures available in the discovery mode. A deeper analysis is underway, which will make use of the faster optimization process described above to carry out more thorough runs in discovery mode. Another relevant topic for future work would be to consider multi-day returns. A brief analysis, again using PS1, showed that k-day returns, with k ranging from 2 to 7, are larger than $\bar{r}_1$ and show a monotonous increase as k goes from 2 to 7. However, the annualized Sharpe ratios are smaller than for the one-day returns.

6 Conclusion

In conclusion, the main result of the work presented above is that candlestick pattern sets can indeed be used for short-term (one-day, in this case) prediction of financial time series, and that the optimization framework presented here is able to find such pattern sets, particularly in the tuning mode, where a user provides a starting point in the form of a loosely defined pattern set. A crucial feature is the possibility of modifying both the structure and the parameters of the patterns sets during optimization. However, the usual caveats of course apply, namely that other data sets may yield different results. Still, the fact that the pattern sets perform well over previously unseen validation data, and even on validation data from a completely different class of stocks (as shown for two of the patterns sets in the discussion above) indicates that those pattern sets indeed capture important aspects of the time series in question.

Acknowledgment. The author would like to thank Prof. Philip Treleaven at University College London, for many interesting discussions.

References

1. Fama, E.F.: The efficient market hypothesis: a review of theory and empirical work. J. Finance **25**, 383–417 (1970)
2. Malkiel, B.G.: Reflections on the efficient market hypothesis: 30 years later. Financ. Rev. **40**, 1–9 (2005)
3. Park, C.-H., Irwin, S.H.: The profitability of technical analysis: a review. Technical Report AgMAS Project Research Report No. 2004–04 (2004)
4. Bajgrowicz, P., Scaillet, O.: Technical trading revisited: false discoveries, persistence tests, and transaction costs. J. Financ. Econ. **106**, 473–491 (2012)
5. Caginalp, G., Laurent, H.: The predictive power of price patterns. Appl. Math. Finance **5**, 181–205 (1998)
6. Lu, T.-H., Shiu, Y.-M., Liu, T.-C.: Profitable candlestick trading strategies—the evidence from a new perspective. Rev. Financ. Econ. **21**, 63–68 (2012)
7. Marshall, B.R., Young, M.R., Cahan, R.: Are candlestick technical trading strategies profitable in the Japanese equity market. Rev. Quant. Finan. Account. **31**, 191–207 (2008)
8. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press (1975)
9. Koza, J.R., Programming, G.: On the Programming of Computers by Means of Natural Selection. MIT Press (1992)
10. Hu, Y., Liu, K., Zhang, X., Su, L., Ngai, E.W.T., Liu, M.: Application of evolutionary computation for rule discovery in stock algorithmic trading: a literature review. Appl. Soft Comput. **36**, 534–551 (2015)