Soft Temporal Patterns for Technical Analysis of Stock Markets

G.K. Palshikar\textsuperscript{*}, Sunod Kumar\textsuperscript{**}, S. Achalia\textsuperscript{+}, P. Alhat\textsuperscript{+}, Y.M. Dhimate\textsuperscript{*}

\textsuperscript{*}Tata Research Development and Design Centre (TRDDC),
54B, Hadapsar Industrial Estate, Pune 411013, India.
email: girishp@pune.tcs.co.in

\textsuperscript{**}Tata Consultancy Services (TCS),
54B, Hadapsar Industrial Estate, Pune 411013, India.
email: sunodk@pune.tcs.co.in

\textsuperscript{+}Department of Computer Engineering,
College of Engineering, Pune 411005, India.
email:{asudeep,pravinalhat,yogeshmd}@indiatimes.com

Abstract. Traders and investors involved in stock market periodically need to take buying and selling decisions (which stocks? when? at what price? what quantity?). The decisions have to be consistent with the most common patterns in the historical movement of stock prices, so as to minimize likely losses and missing possible opportunities. Technical analysis is a set of practical techniques in which most such common patterns and their impact on the decisions are specified. The patterns are temporal, composed from more primitive patterns and are soft in the sense that their actual “instances” are not identical to the ideal patterns. We describe a tool for stock market users to enable them to base their decisions on knowledge of technical analysis. Users can define their own soft temporal patterns and the tool automatically detects their instances in the given timestamped stock market databases. The pattern specification language is based on a fuzzy temporal logic and a fuzzy version of Allen’s interval algebra. Pattern detection is based on the idea of computing degree of truth of a given temporal formula in the given temporal interpretation obtained from the database.

Keywords: Fuzzy logic, Temporal logic, Interval algebra, Pattern recognition, Stock markets, Technical analysis.

1. Introduction

Financial domains offer challenging and complex real-life problems where AI techniques can find rewarding applications. Stock market in particular is an intriguing domain: observations of the stock market are precise (numeric temporal data about prices, volumes, trading indexes such as BSE Sensex) but the underlying processes that affect the market are not well understood. A large number of competing traders participate in trading. Stock prices and trading volumes are affected by many interdependent factors: demand and supply, company specific business conditions, banking factors such as interest rates, economic conditions, political events and even
environmental and mass psychology factors (“market sentiments”, “selling tendency”, “investor confidence” etc.). The stock market domain contains difficult problems for decision-making, forecasting, optimization, stochastic system modeling etc. Many different AI techniques have been applied to various problems in the stock market domain: expert systems, fuzzy logic [3], neural networks [13], genetic algorithms [2], fractals, data mining and machine learning etc. IEEE even holds an annual conference *Computational Intelligence for Financial Engineering (CIFER)*.

We look at a generic problem in stock market that involves activities of a small investor or trader who maintains a portfolio containing shares of some companies. As input, the investor has available to him a daily summary of the trading at a particular stock exchange (Table 1). The investor periodically needs to take buying and/or selling decisions so as to maximize the returns and minimize losses: which stocks? when? at what price? what quantity?. The decisions have to be consistent with the most common patterns in the historical movement of stock prices, so as to minimize likely losses and missing possible opportunities.

Table 1. A part of the data summarizing the trading in NSE on a particular day.

<table>
<thead>
<tr>
<th>symbol</th>
<th>open</th>
<th>high</th>
<th>low</th>
<th>close</th>
<th>value</th>
<th>quantity</th>
<th>trades</th>
<th>52whigh</th>
<th>52wlow</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABB</td>
<td>267.1</td>
<td>271.95</td>
<td>262.15</td>
<td>263.4</td>
<td>2698349.15</td>
<td>10186</td>
<td>207</td>
<td>292</td>
<td>173</td>
</tr>
<tr>
<td>ACC</td>
<td>160</td>
<td>160.25</td>
<td>157.5</td>
<td>158</td>
<td>82488294.95</td>
<td>519250</td>
<td>4368</td>
<td>18</td>
<td>0.65</td>
</tr>
<tr>
<td>BHEL</td>
<td>171.8</td>
<td>172</td>
<td>167.1</td>
<td>168.25</td>
<td>40843416.9</td>
<td>240663</td>
<td>1970</td>
<td>196.4</td>
<td>104.1</td>
</tr>
</tbody>
</table>

The trading community in stock market has evolved two major approaches for this problem [4,5]. In *fundamental analysis (FA)*, the financial status (health) of a company is adjudged (e.g., by computing various ratios), based on its balance sheets, profit and loss statements, analysis of the competition, government policies, economic situations etc. Based on this analysis, the trader hopes to find the intrinsic value of the company’s stock and use it for early spotting of opportunities or risks; e.g., if the intrinsic price is higher than current price, this may indicate a signal to buy the stock. Short-term fluctuations in prices and volumes are mostly ignored.

In *technical analysis (TA)*, various graphs (or charts) depicting the historical progress of stock prices, market averages, etc. are studies to find various trends in supply and demand and to identify gradual shifts in or points of departure for these trends. TA believes that all factors that affect the market are summarized in the current price and volume figures; thus any further explicit analysis of the external factors that affect the market is largely disregarded. While debates rage on about the merits of FA and TA and their comparison with other approaches such as those based on statistical models, neural networks etc., in practice, TA has a large following among traders and investors. Many commercial tools such as MetaStock (www.equis.com), Advanced GET (www.tradingtech.com), Omega TradeStation (www.trade-station.com) etc. offer support for (mostly manual) decision making based on TA.

Understanding charts and identifying trends is an extremely important part of TA. Unfortunately, this task is almost entirely manual and it is performed visually by TA experts (and therefore it is time consuming and error-prone). In this paper, we report
theoretical technique and based on it a software tool we have built to support TA as follows. The users define specific soft temporal patterns of their interest. The tool then searches the database to automatically identify and report instances where the given patterns have occurred. A soft temporal pattern is essentially a high-level summary of vast temporal data. We have often observed that users organize patterns hierarchically and compose more complex patterns using logical and temporal connectives and given domain-specific primitive patterns or features. A temporal pattern is qualitative in the sense that one does not specify actual time instants and intervals but deals only with temporal relationships. It is approximate in that the statements may have fuzzy (rather than Boolean) truth-values. Examples of such soft temporal patterns are:

- whenever the price of a stock is very high or very low, trading in it is rather low
- large buying of a stock within a small time interval leads to a sharp rise in the price
- as the market price of a stock increases, so does the volume traded
- selling tendency increases whenever stock price goes above the previous maximum

Exceptions to such normal patterns can be termed as alarms. Early warnings can be similarly defined in terms of patterns. See [8,9] for some related work. In [12], we have applied similar pattern detection techniques for detection of long-term frauds in stock market trading.

In this paper, we study the specific types of charts and patterns that TA users require. The rest of the paper is organized as follows. Section 2 discusses TA and the charts and patterns in TA. Section 3 outlines how soft temporal patterns needed in TA are defined using a fuzzy temporal logic and how instances of such a pattern are detected in a relational database. Section 4 reports the design and implementation of a tool for TA based on our approach. Section 5 includes conclusions and further work.

2. Technical Analysis of Stock Market Databases

Technical analysis (TA) believes that the current price and trading volume of a stock summarize (and are the result of) all the important business and market factors. Hence technical analysts draw a number of visual charts depicting historical progress of various observables such as in price and volume of a stock, market indices, market averages and many other statistical indicators. Figure 1 shows a chart that plots the daily closing price of a stock and also marks some trends and signals on the chart.

Technical analysts have a repository of a large number of commonly occurring trends and patterns. Examples: bump and run, double top, double bottom, head and shoulders top, head and shoulders bottom, falling wedge, rising wedge, rounding bottom, triple top, triple bottom, cup with handle, flag, pennant, symmetric triangle, ascending triangle, descending triangle, price channel, rectangle, measured bear move, measured bull move etc. To a technical analyst, an occurrence of a particular pattern in a chart indicates a specific signal. Technical analysts visually analyze the
charts to detect occurrences of the various trends and patterns from these repositories and take actions (such as buy or sell a specific stock) based on these occurrences.

As an example, Figure 2(a) shows an ideal head and shoulder top pattern that technical analysts look for in the price chart of a stock. The pattern is ideal in the sense that the curves in an actual chart will not be so smooth (will contain “wiggles” or “whipsaws”), the neckline will not exactly touch the base points of the shoulders and so on. This pattern essentially consists of a sequence of 3 adjacent “inverted cups” (left shoulder, head and right shoulder) such that the straight-line (i.e., the neckline) tangent to the base of the two shoulders has nearly horizontal. The breakout point, where the actual price falls below the neckline indicates a sell signal.

Figure 2(b) shows an idealized bump and run reversal (BARR) pattern in the price chart of a stock. In the first lead-in part that lasts for 1 month or more, the price increases in an orderly manner (the trend-line is moderately steep). Next comes the bump i.e., a steep increase in the price, indicative of excessive speculation. The bump does not last too long. Towards the end of the bump, there is a bump rollover, a small flat top, which may sometimes include 1 or 2 small peaks. In the next run phase, the price starts to fall rapidly and continues to fall further after “breaking” the trend-line. This pattern is indicative of wild (but unsustainable) speculations that drive up the price. Most patterns are also correlated with the trading volumes; e.g., the volume is neither too high nor too low in the lead-in phase of the BARR pattern. The technical analysts also impose a minimum/maximum or typical duration requirements on the pattern as a whole or on parts of the pattern. Technical analysts also look for well-defined confirmations for a pattern; e.g., for BARR, a typical confirmation that the run phase has reached the climax is that volume reaches a steady level and price develops 1 or 2 small peaks.

Note that the patterns are composed from sub-patterns (e.g., head and each shoulder are sub-patterns); the patterns involve temporal relationships and the pattern instances are not identical to the idealized pattern. The key question we address and solve is: can such soft temporal patterns be pre-defined (specified) in a simple manner and detected efficiently in the given database? We propose a fuzzy temporal logic as a specification notation for soft temporal patterns and show how each of these patterns can be defined as a formula in this logic. The database defines a particular
interpretation for the given formula in this logic and the pattern detection problem is posed and efficiently solved as computing the truth-value of a given formula in a given interpretation.

(a) Head and shoulder top pattern (b) Bump and run reversal pattern

Fig. 2. Some idealized patterns in the price chart of a stock.

3. Soft Temporal Patterns

We define a fuzzy propositional temporal logic (FzPLTL), which is a standard propositional fuzzy logic with the addition of usual temporal modalities adapted for fuzzy applications. We propose a view that such a fuzzy temporal notation is appropriate for the specification of soft temporal patterns in temporal databases. Domain-specific features are used to define the primitive fuzzy temporal propositions. These fuzzy temporal propositions are then used to compose complex patterns using standard logical and temporal connectives. Thus, in this paper, a temporal pattern is actually a formula in FzPLTL. Temporal logic, like FzPLTL, abstracts away the details of actual time instants/periods and allows one to focus on inter-relationships between events. It provides a rich variety of temporal and logical connectives for pattern composition. Fuzziness in the notation allows specification of inexact patterns.

3.1 Non-temporal Fuzzy Patterns

A number of primitive features can be defined in such a system. A primitive feature is a function from timestamp (and possibly other parameters) to a set of numbers. For the database in Table 1, the primitive features:

- price : TIMESTAMP × SYMBOL → \n
- volume : TIMESTAMP × SYMBOL → \n
- moving_avg_price : TIMESTAMP × SYMBOL → \n
- moving_avg_volume : TIMESTAMP × SYMBOL → \n
respectively return the price, volume and (appropriately computed) moving averages of the price and volume for the given company at various timestamps; SYMBOL and TIMESTAMP are respectively the set of values for the company IDs and timestamps that occur in database of Table 1. We assume that TIMESTAMP denotes a finite linear discrete timeline consisting of an ordered sequence of (not necessarily equally spaced) time instants. Note that each feature is based on a continuity assumption, which specifies the values at timestamps where they cannot be directly computed. For example, if there are no transactions at timestamp t for a specific company, then for the feature price, we can either return (a) the price used in the last transaction (b) the average price of the last and the next transaction (c) return a special failure value like say 0.0 or –1.0 and so on. A fuzzy feature is a feature whose return value is a real number in the closed interval [0, 1]. For example:

\[
\begin{align*}
\text{price}_{\text{high}}, \text{price}_{\text{low}} &: \text{TIMESTAMP} \times \text{SYMBOL} \to [0,1] \\
\text{price}_{\text{increasing}}, \text{price}_{\text{decreasing}}, \text{price}_{\text{steady}} &: \text{TIMESTAMP} \times \text{SYMBOL} \to [0,1] \\
\text{volume}_{\text{high}}, \text{volume}_{\text{low}}, \text{volume}_{\text{steady}} &: \text{TIMESTAMP} \times \text{SYMBOL} \to [0,1] \\
\text{volume}_{\text{increasing}}, \text{volume}_{\text{decreasing}} &: \text{TIMESTAMP} \times \text{SYMBOL} \to [0,1]
\end{align*}
\]

The value v of the fuzzy feature price_{high} for a given company c and a given timestamp t denotes the degree of truth of the proposition (or statement) "price of company c is high at time t". Similar fuzzy features can be defined for volume. It is important to note that each fuzzy feature is in some sense a parameterised fuzzy set. These fuzzy features are computed using standard fuzzy set membership functions. Now one can easily form fuzzy expressions using the usual fuzzy logical connectives ¬ (not), ∧ (and), ∨ (or), → (implies) etc. Each such expression has a fuzzy degree of truth, which is a real number in the closed interval [0,1]. We use relational expressions involving non-fuzzy primitive features and fuzzy features as building blocks (i.e., fuzzy propositions). Obviously, Boolean expressions (which have the degree of truth as 0 or 1 only) are special cases of the fuzzy expressions; e.g., \(\text{price}(101) < 35.5\) is a Boolean expression either true or false at any instant. We omit the timestamp argument from fuzzy expressions to indicate that the fuzzy expression can be evaluated at different times. For example,

1. prices of companies ABB and BPL are nearly equal
   \(\text{price}('\text{ABB}') \approx \text{price}('\text{BPL}')\)
2. if the price of a stock ABB is very high or very low, volume is fairly low
   \((\text{very price}_{\text{high}}('\text{ABB}') \lor \text{very price}_{\text{low}}('\text{ABB}')) \rightarrow \text{fairly volume}_{\text{low}}('\text{ABB}')\)
3. Whenever the price approaches the 52-week high, the volume is high
   \(\text{price}('\text{ABB}') = \text{price}_{\text{52weekhigh}}('\text{ABB}') \rightarrow \text{high volume}_{\text{high}}('\text{ABB}')\)
4. Price of a company ABB is steady
   \(\neg \text{price}_{\text{increasing}}('\text{ABB}') \land \neg \text{price}_{\text{decreasing}}('\text{ABB}')\)

### 3.2 Temporal Fuzzy Patterns

Table 2 lists some future temporal connectives in the standard linear temporal logic. Past connectives can be similarly defined; e.g., \(\bigcirc\), \(\square\) denote at previous instant and always in the past. We have defined fuzzy semantics for these connectives [10].
Table 2. Some future time connectives in FzPLTL.

<table>
<thead>
<tr>
<th>Connective</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X X</td>
<td>X is true at the next instant</td>
</tr>
<tr>
<td>X</td>
<td>X is true now and for the next n time instants after now</td>
</tr>
<tr>
<td>X</td>
<td>X is true now and at all future time instants</td>
</tr>
<tr>
<td>X</td>
<td>X is true on the average at now and for the next n time instants after now</td>
</tr>
<tr>
<td>X</td>
<td>X is true on the average at now and at all future time instants</td>
</tr>
<tr>
<td>X</td>
<td>X is true either now or at some future time instant</td>
</tr>
<tr>
<td>X</td>
<td>X is true now or within at most next n time instants</td>
</tr>
<tr>
<td>X U Y</td>
<td>Y is eventually true at some time instant from now onwards and X is true everywhere from now up to that time instant</td>
</tr>
</tbody>
</table>

Now we are in a position to define fuzzy temporal expressions (i.e., patterns) using both the fuzzy logical as well as fuzzy temporal connectives and using the given features as building blocks. We assume the existence of suitable features and truth modifiers to construct these patterns. Note that some of these patterns are written better using the past temporal connectives. Obviously, there is no unique way of writing a pattern. Examples of fuzzy temporal patterns are:

5. If the price is very high and increasing then it will become steady within 5 days
   \[ \text{very price\_high('ABB')} \land \text{price\_increasing('ABB')} \rightarrow \Diamond_5 \text{price\_steady('ABB')} \]

6. Price remains steady until the volume starts increasing
   \[ \text{price\_steady('ABB')} U \text{volume\_increasing('ABB')} \]

The truth-value of a FzPLTL formula is computed at each instant in the time domain [10]. Figure 3 shows how the truth-value of the formula (2) from section 3.1 varies over the instances in the time domain. Note that the formula has a very high truth-value (> 0.9) at most of the time; thus the pattern (2), which indicates a characteristic normal trading behaviour, is strongly true for the company ABB.

### 3.3 Interval Algebra Facilities

In many TA patterns, we are concerned with qualitative relationships among the time intervals where various sub-patterns have a strong presence. Table 3 shows the 13 possible relationships between 2 intervals, as defined in Allen’s interval algebra. We
provide meta-logical facilities to extract intervals during which a given FzPLTL formula is strongly true and also provide suitably fuzzified versions of Allen’s interval algebra, as part of the pattern specification notation \[11\].

We model a \emph{closed time interval} as a tuple \([t_i, t_j]\) as a sub-sequence of the given timeline and say that it consists of the time points \(t_i, t_{i+1}, \ldots, t_j\). We introduce the expression \([F]_\alpha\) to denote the maximally extended time interval where a fuzzy temporal formula \(F\) has the average degree of truth above the given threshold \(\alpha\). That is, if \([F]_\alpha = I = [t_i, t_j]\) then \(\mu(F, I) = ([TV(F) at t_i] + TV(F) at t_{i+1} + \ldots + TV(F) at t_j]) / n \geq \alpha\), where \(n\) is the number of time instants in the interval \([t_i, t_j]\). The interval \([F]_\alpha\) is \emph{maximally extended} in the sense that no time instant can be added to it (say \(t_{j+1}\)) without reducing the average value of \(F\) over the extended to interval to less than \(\alpha\). When the truth-value of a pattern over time is depicted as in Figure 3, one often notices that the pattern is significantly present in several intervals. Concept of such an interval is exactly captured by a maximally extended interval.

Given a time domain \(\text{TIMESTAMP}\), a threshold \(\alpha \in [0, 1]\), an FzPLTL formula \(F\) and a valuation function \(TV(F) : \text{TIMESTAMP} \rightarrow [0,1]\), we can obtain a sequence of non-overlapping intervals \(I_i = \langle I_i, I_{i+1}, \ldots, I_k \rangle, k \geq 1\), such that (i) \(\mu(F, I_i) \geq \alpha\) for all intervals \(I_i\) in \(I_i\), \(1 \leq i \leq k\); and (ii) each interval \(I_i\) in \(I_i\), \(1 \leq i \leq k\), is maximally extended; and (iii) \(I_{i+1}\) is after \(I_i\) for all \(1 \leq i \leq k-1\). Such a sequence of intervals \(I_i\) is called an \emph{occurrence} of \(F\) in \(\text{TIMESTAMP}\). In general, occurrence of \(F\) is not unique. We have published an algorithm \[10\] to extract the occurrence (i.e., a sequence of all maximally extended intervals) for a given FzPLTL formula \(F\).

<table>
<thead>
<tr>
<th>Relation</th>
<th>Example</th>
<th>Inverse Relation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>X before Y</td>
<td>([20, 30])</td>
<td>X after Y</td>
<td>([40, 60])</td>
</tr>
<tr>
<td>X meets Y</td>
<td>([20, 30]) meets ([31, 50])</td>
<td>X met_by Y</td>
<td>([31, 50]) met_by ([20, 30])</td>
</tr>
<tr>
<td>X overlaps Y</td>
<td>([20, 30]) overlaps ([25, 40])</td>
<td>X overlapped_by Y</td>
<td>([25, 40]) overlapped_by ([20, 30])</td>
</tr>
<tr>
<td>X during Y</td>
<td>([20, 30]) during ([10, 50])</td>
<td>X contains Y</td>
<td>([10, 50]) contains ([20, 30])</td>
</tr>
<tr>
<td>X starts Y</td>
<td>([20, 30]) starts ([20, 50])</td>
<td>X started_by Y</td>
<td>([20, 50]) started_by ([20, 30])</td>
</tr>
<tr>
<td>X finishes Y</td>
<td>([20, 30]) finishes ([10, 30])</td>
<td>X finished_by Y</td>
<td>([10, 30]) finished_by ([20, 30])</td>
</tr>
<tr>
<td>X equals Y</td>
<td>([20, 30]) equals ([20, 30])</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.4 TA Patterns as FzPLTL Formulae

It is easy to see how each soft temporal pattern for TA described earlier can be represented as a formula in FzPLTL. For example, a shoulder can be modeled as:
\[
[[\text{price_increasing(}‘\text{ABB}\text{’)}]_{0.6} \text{ fz_meets } [[\text{price_decreasing(}‘\text{ABB}\text{’)}]_{0.6}
\]
where \text{fz_meets} is the fuzzy interval algebra connective corresponding to meets. The entire head and shoulder top pattern can be similarly modeled. We have implemented a general-purpose pattern engine called SNIFFER in which users can define their own soft temporal patterns as FzPLTL formulæ. The tool searches for and reports the occurrences of a given pattern in given temporal database. We now report how SNIFFER has been tailored for patterns in TA.
4. The TradeWorX Tool

We have developed a software tool that includes a support for automatic detection of many of the well-known soft temporal patterns used by TA. The user can select the pattern to be detected, provide values for certain parameters and the tool detects and reports the occurrences, if any, of the given pattern in the given stock market database. Table 1 shows the format of the relational database in which TA patterns are detected by the tool. Note that this is a temporal database, because each record contains a timestamp field. The database is currently in MS-ACCESS. The tool is implemented using Prolog and Visual C++ on MS-Windows. The user interacts with the underlying pattern detection engine SNIFFER through a graphical user interface (GUI). The first screen of the GUI requires the user to input a number of parameters like the name of the script (symbol) and the name of the pattern to be detected in that script. The GUI also asks for several other parameters (MaxIncR, MinIncR, MaxDecR, MinDecR, MeetMin, MeetMax, Threshold, Separation, MinInt Size and Equal Extent); a discussion of these parameters is omitted here for brevity. Figure 4 asks the tool to detect the head and shoulders top pattern in the stock ABB. Values for
parameter as shown in Figure 4 work for most scripts but occasionally a change in certain parameter values may be required depending on the script selected by the user. Figure 4 also shows the output generated by the tool for the parameters shown. Note that the detected head and shoulder pattern is quite different from the idealized one in Figure 2(a). Figure 5 shows the bump and run reversal pattern detected by the tool.

5. Conclusions and Further Work

We are investigating a number of further aspects of this research. First, we wish to enrich our library of patterns in TA automatically detected by the tool. We are also planning to add many more features (such as automated calculations of buy and sell signals and alerts, and optimization techniques for portfolio management) so as to make the tool commercially viable. Next, we are looking into the possibility of automatically calculating appropriate values of the control parameters for a pattern (these are currently specified by the user). We are also investigating applications of these techniques in other fields such as defence, diagnosis, risk analysis, resource management and performance monitoring.

Acknowledgements: We thank Prof. Mathai Joseph, Executive Director, TRDDC, for his support and colleagues in TCS for help. The first authors thanks Dr. Manasee Palshikar.

References