Decision Support Systems 54 (2013) 838–845

Contents lists available at SciVerse ScienceDirect

Decision Support Systems

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

Trading team composition for the intraday multistock market☆,☆☆

Leandro G.M. Alvim *, Ruy L. Milidiú

Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Brazil

ARTICLE INFO

Article history:
Received 19 April 2011
Received in revised form 29 June 2012
Accepted 18 September 2012
Available online 25 September 2012

Keywords:
Trading team
Partial Least Squares
Weighted Interval Scheduling
Computational finance
Machine learning

ABSTRACT

Automated traders operate market shares without human intervention. We propose a Trading Team based on atomic traders with opportunity detectors and simple effectors. The detectors signalize trading opportunities. For each trading signal, the effectors follow deterministic rules on when and what to trade in the market. The detectors are based on Partial Least Squares. We perform some trading experiments with twelve BM&FBovespa stocks. The empirical findings indicate that the proposed trading strategy reaches a 77.26% annualized profit, outperforming by 380.07% the chosen baseline strategy with a 16.07% profit. We also investigate Multistock Resolution Strategy (MSR) performance subject to brokerage commissions and income tax. Whenever the initial investment is at least US$ 50,000, the MSR strategy provides a profit of at least 38.63%.

1. Introduction

Automated traders are artifacts that operate market shares without human intervention aiming to maximize the investor's earnings. Recently, several studies have been conducted using a combination of machine learning algorithms for market forecasting and automated traders [1,2,5,8–11,16,18,20,21,26,37]. Despite these efforts to, accurately, predict future stock trends and to develop trading strategies that turn good predictions into profits are still two major challenges.

Here, we propose a novel architecture that automatically selects stocks and trading times for a market day. The trading architecture contains trading opportunity detectors and simple trade effectors. The detectors task is to signalize trading opportunities. Given a specific trading signal, the effectors follow deterministic rules on when and what to trade in the market. For the trading task, we build a trading team by combining atomic traders that operate with different stocks and time resolutions, with the help of several trading opportunity detectors. The detectors are based on Partial Least Squares (PLS), whereas the team is selected by maximizing the investor return over a market operation dataset.

We build several intraday traders, each one with a corresponding (stock, operation window) pair. We evaluate these traders by their corresponding trading returns. The trader team return is given by the composition of its selected trader rewards.

We perform some trading experiments with twelve BM&FBovespa[1] stocks. The empirical findings are shown in Table 1. The results indicate that the Multi Stock-Resolution strategy outperforms a chosen baseline on daily average profit. Since there is no standard error overlap, the MSR average profit is statistically different from BLS average profit. We also investigate MSR performance subject to brokerage commissions and income tax. Whenever the initial investment is at least US$ 50,000, the MSR strategy provides a profit of at least 38.63%.

The main contribution is an automatic portfolio selection architecture for a trading day. The proposed architecture combines an optimization module with several machine learning opportunity detectors.

This work is organized as follows. In Section 2, we describe the trading scenario and its corresponding assumptions. In Section 3, we describe the PLS detectors and its corresponding quality metrics, that we use to forecast trading opportunities. In Section 4, we investigate atomic traders and trader teams. Additionally, we formulate the Trader Team Composition problem and show its solution. In Section 5, we describe the baseline trader that uses a classical trading approach. In Section 6, we show the empirical evaluation, that we use to assess the proposed strategy performance. Moreover, we compare the multistock trading strategy findings with the baseline. Finally, in Section 7, we present the conclusions.

2. Trading scenario

2.1. Market assumptions

For mathematical modeling simplicity, we make the following market assumptions [7]. The *all or nothing* trade position at all times is either entirely bond or entirely stock.

☆ Partially supported by CNPQ 141281/2009-0.
☆☆ Partially supported by FAPERJ project E-26/170.028/2008.
* Corresponding author at: Rua Marquês de São Vicente, 225, Gávea, Rio de Janeiro, RJ, 22451-900, Brazil. Tel.: +55 2181734381.
E-mail addresses: lalvim@inf.puc-rio.br (L.G.M. Alvim), milidiu@inf.puc-rio.br (R.L. Milidiú).

[1] BM&Fovespa is the official Brazilian stock market exchange hosted in São Paulo.

0167-9236/$ – see front matter
http://dx.doi.org/10.1016/j.dss.2012.09.009

Table 1
Strategies comparison.

Strategy	Avg. daily profit (%)	Std. error (%)
Baseline	0.06	0.036
Multi stock-resolution	0.24	0.039

All or nothing is a common approach regarding maximizing profits for short-term strategies [7,15,35]. Another approach is portfolio selection, which diversifies a portfolio in order to reduce the risk over time. This approach is a standard for long-term strategies [24,28,39] and we point a research direction in our conclusions. Here, we adopt the first one.

The *all or nothing* premise is a greedy heuristic that prioritizes maximizing profit over time. To reduce the trading risk, we automatically select the thresholds for each trader, described in Section 2. The thresholds represent safe distances from a loss transaction.

Fractional market arbitrary amounts of stock or bond can be bought or sold at any time. *No market impact* trades can be placed without affecting the quoted price. Market impact is not significant when dealing with moderate and high liquidity stocks, what is our case. Moreover, simulating market impact is not a trivial task and still a research topic.

2.2. Trading costs

To reproduce a more realistic simulation scenario, we consider the following costs [15]: brokerage commissions and income tax.

Brokerage commission is a fee charged by the financial intermediary institutions to its customers. For each operation, we consider a typical US$11.00 Brazilian day trade brokerage commission [15]. Regarding Brazilian day trading rules, we deduct 20% of the earnings.

2.3. Performance metrics

To evaluate trading strategies, we apply the usual actual trading metrics. We divide the metrics into four modules: return, risk, gain per risk and trading. Each module contains performance indexes.

The return module indexes are daily average profit, minimum and maximum profit performances.

The risk module indexes consist of two risk metrics: Maximum Draw Down [23] and Ulcer Index [25]. The Maximum Draw Down is a measure of the maximum decline from a peak to a bottom performance in a specific interval. A high value on a profit series represents a high loss. The Ulcer Index evaluates drawn down depths and durations. Opposed to the standard deviation that measures risk as downward and upward profit moves, Ulcer Index associates risk only to downside profit moves.

Another is the gain per risk module. This module includes the gain per unit risk metrics [6,33] namely Sharpe ratio, Martin ratio, Calmar ratio, Sterling ratio and Burke ratio. Sharpe ratio is the excess return divided by the standard deviation of returns. Martin ratio is the investment excess return divided by Ulcer Index risk. Calmar ratio is the excess return divided by the maximum drawdown. Sterling ratio is the excess return divided by the averaged k-largest drawdowns. Burke ratio is the excess return divided by the square root of the sum of the squared k-largest drawdowns. For Sterling and Burke ratios, we set $k=3$ [33].

Finally, the trading module indexes that measure the aggressiveness of our strategy. We choose some indexes such as winning trades, losing trades, average number of trades, average gain divided by average loss.

3. Trading opportunity detectors

Stock market forecasting is a challenging topic for both investors and researchers. This is because the stock market data suffers from non-linearity and uncertainty [2,8,20]. Standard statistical forecasting strategies provide a partial solution to these issues. Thus, stock market forecasting asks for alternative methods, such as Artificial Neural Networks (ANN's) [17,18,26,30,32,34], Support Vector Machines (SVM's) [10,16,21], Support Vector Regression (SVR) and Partial Least Squares (PLS) [5] and Decision Trees [36].

Our approach uses machine learning. Next, we define the learning task, the learning algorithm, the feature engineering, the performance metrics and modeling.

3.1. The task

We say that there is a *trading opportunity* of stock s at time t whenever we forecast a significant price rise of s at time $t+\delta$. If that is the case, a simple strategy is to buy s at time t and sell it at time $t+\delta$. Hence, we simplify the price forecasting problem to a simple trading opportunity detection task. Moreover, we move from a regression to a classification task.

Therefore, we build a set of machine learning based opportunity detectors, aimed at forecasting future price rises.

3.2. Detectors

We formulate trading opportunity detection as a binary classification task. Thus, we classify every time instant t either as a trading instant or not. We solve this problem by applying a supervised learning algorithm. In order to build the required training set, we just examine each past instant and check whether it is a trading opportunity or not.

We use a two step scheme to build the detectors. First, we use regression to provide effective intraday price forecasts. Next, we discretize these forecasts into *up* and *down* price trends. We define an up trend as relative price increase of more then , where is a chosen threshold. Otherwise, we say that we have a down trend. To choose the threshold we apply a grid search heuristic. We describe the details in the Modeling subsection.

For the price forecasting step, we apply Partial Least Squares algorithm based on [31] study. We choose PLS because there is only one parameter to adjust and it is computationally faster than support vector regression (SVR) and Artificial Neural Networks algorithms. Moreover, [5] PLS results indicate competitive performance against SVR for volume forecasting.

3.3. Feature engineering

By examining past market information, technical analysts try to explore patterns that would help to forecast future market opportunities. Despite its theoretical advances, researchers do not find them effective [3,14]. This fact led to the widespread support of the Efficient Market Hypothesis. According to [13], what caused technical principles to fail in the 60s were *ad hoc* specifications of trading rules that led to data snooping.

Instead of using trading rules based on the technical indicators, machine learning algorithms are being used to learn trading rules [10,22,29] from a set of technical analysis indicators.

In Appendix A, we enumerate the technical indicators that we use to improve the proposed detector input features set. They capture the following concepts: trend, risk, volume and momentum. Our goal here is to provide as much market information to our predictors as possible. We also apply simple price and volume features such as opening, closing, maximum and minimum.

Since our problem incorporates sequential data, we apply a sliding window method (SWM) [12] to our features. The objective of the SWM is to convert a sequential supervised learning problem into a classical supervised learning problem.

Suppose we have an ordered data set $D=\{(x_t,y_t)\}_{t=1}^{n}$ composed of n samples, where each pair (x_t,y_t) contains the input attribute vector x_t and the output variable y_t at time t. In the SWM approach, one maps a w-delayed input feature set $x_{i-w},\ldots,x_{i-2},x_{i-1}$ to the corresponding