Pattern Recognition Letters 97 (2017) 21–28

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Using real-time cluster configurations of streaming asynchronous features as online state descriptors in financial markets

Dieter Hendricks*

School of Computer Science and Applied Mathematics, University of the Witwatersrand, WITS 2050, Johannesburg, South Africa

ARTICLE INFO

Article history:
Received 22 March 2016
Available online 27 June 2017

MSC:
41A05
41A10
65D05
65D17

Keywords:
Online learning
State space discovery
Asynchronous data
Financial markets

ABSTRACT

We present a scheme for online, unsupervised state discovery and detection from streaming, multi-featured, asynchronous data in high-frequency financial markets. Online feature correlations are computed using an unbiased, lossless Fourier estimator. A high-speed maximum likelihood clustering algorithm is then used to find the feature cluster configuration which best explains the structure in the correlation matrix. We conjecture that this feature configuration is a candidate descriptor for the temporal state of the system. Using a simple cluster configuration similarity metric, we are able to enumerate the state space based on prevailing feature configurations. The proposed state representation removes the need for human-driven data pre-processing for state attribute specification, allowing a learning agent to find structure in streaming data, discern changes in the system, enumerate its perceived state space and learn suitable action-selection policies.

1. Introduction

Machine learning has become ubiquitous in high-frequency financial markets, as technological advances enable low-latency automated algorithms to replace functions traditionally performed by human traders, portfolio managers, risk managers and regulators. This is particularly true for trading algorithms, where reinforcement learning algorithms have recently been considered as dynamic alternatives to traditional stochastic control techniques (such as those found in [1,3,7,11,12]) for mapping optimal trajectories through the system. Nevmyvaka [30], Nevmyvaka et al. [31] were among the first authors to consider a reinforcement learning agent for optimal limit order placement for a liquidation program. They used a discrete-state, discrete-action *Q-learning* agent which converged to a policy for the optimal price at which to post the remaining inventory in the market, based on the time remaining in the liquidation program, remaining inventory to trade and domain-informed public state attributes, such as prevailing spreads, price levels and volumes. Hendricks and Wilcox [21] considered a similar problem, demonstrating that a reinforcement learning agent can be used to adapt a trading strategy with respect to prevailing spread and volume dynamics, executing a sequence of optimised market orders. These authors demonstrate a significant positive improvement on cost of trading compared to state-of-the-art techniques, motivating reinforcement learning as a suitable framework for online learning agents in financial markets. Both studies, however, use a subjective set of attributes for state representation in the learning algorithm. While the choice is informed by domain knowledge and may be suitable at the operating scale of a human trader, we conjecture that a more objective representation may yield better trading policies for agents operating at machine scale.

It is well known that the performance of certain classes of machine learning algorithms is strongly dependent on the choice of data representation, or features, upon which they are applied ([2,22,26]). This is likely due to certain forms of representation masking exploitable characteristics explaining variations in the data, or at least burying them in layers which cannot be detected by the learning algorithm. As such, significant effort can be spent on data pre-processing, using domain knowledge to inform appropriate representations for effective machine learning. While such human intervention can be useful to guide learning agents in new domains, it does restrict the agent's discoverable policies to those which mimic policies acceptable to an intuitive *human* agent in the domain. Bengio et al. [2] state that an artificial intelligence should fundamentally understand the world around us, and thus be able to identify and disentangle explanatory features from low-level sensory data without human intervention. In this way, a machine learning agent can provide more general, and sometimes com-

* Current address: Machine Learning Research Group, Oxford-Man Institute of Quantitative Finance, University of Oxford.

E-mail address: dieter.hendricks@eng.ox.ac.uk

http://dx.doi.org/10.1016/j.patrec.2017.06.026

plementary optimal policies to those expected by human agents, thereby cultivating its own distinct *machine intelligence*.

This goal has been recognised by the machine learning community, with a recent surge in scientific activity concerning unsupervised feature learning (or deep learning), seeking the discovery of useful representations which result in more meaningful classifiers and predictors in various domains (see [6,9,10,15,23,25] for some state-of-the-art examples). At a recent NIPS workshop, Mnih et al. presented the first deep learning model to successfully learn control policies from high dimensional sensory data using reinforcement learning [29]. The agent was able to learn to play several Atari2600 games, using a convolutional neural network (CNN) trained using a *Q-learning* algorithm, with only raw pixels as the input. While this is a somewhat different domain to the optimal trade execution problem, it does present certain analogues consistent with our goal: using low-level sensory data (*pixels* here, vs *streaming tick data* for our problem), the CNN is able to abstract useful representations from the raw data and train a *Q-learning* agent to achieve some goal. While it would seem appropriate to apply this technique to our problem, the computational burden of the CNN may be too onerous for our goal of an online near-real-time algorithm using modest hardware. Even recent work of [8] on state-of-the-art, computationally efficient, GPU-optimised CNNs yield computation times of the order of minutes for relatively simple problems.

We are thus tasked with developing a form of state representation which can be constructed directly from raw asynchronous tick data, is able to capture salient features of the limit order book, is computationally efficient for near-real-time use (of the order of seconds) and can be successfully combined with *Q-learning* for optimal trade execution policies. In the following sections, we describe our approach which is able to construct a rich state representation in a computation time of the order of seconds, using relatively modest hardware, enabling near-real-time state detection for online learning.

2. Cluster configurations as temporal state descriptors

In the previous studies considering state representations for high-frequency financial markets, the following pre-processed attributes were used as candidate descriptors: *bid/ask spread, quote volumes, quote volume imbalance, trade price levels* and *traded volumes* ([21,31]). These were informed by common notions and intuition from human traders regarding the typical drivers of trade execution cost when interacting with financial markets. Consider a trader who is only able to execute *market orders* to satisfy an *arrival price objective* in a *limit order book* market. The *limit order book* is a schedule of quoted prices and volumes at which market participants are willing to transact, where *ask quotes* refer to *sell orders* and *bid quotes* refer to *buy orders*, with *bid quote prices* strictly less than *ask quote prices*. Participants who place limit orders effectively achieve can achieve a favourable price if they are willing to wait for the market to move in the direction of their limit price. There is thus uncertainty in the time at which the transaction will take place, if at all. Alternatively, a trader may place a *market order*, whereby they guarantee execution by matching against prevailing limit orders in the system, i.e. a *buy market order* will match against commensurate *limit ask quotes*, with the trade price calculated as the volume-weighted price of the matched ask quotes. The cost of this timing guarantee is thus paying a higher (lower) price for a buy (sell) order, where the extent of this cost is governed by the prevailing *quote depth* in the limit order book. The difference between the highest bid quote price and the lowest ask quote price is referred to as the *spread*.

Given the objective of minimising trading cost with respect to an arrival price benchmark, and the constraint of only executing market orders, a rational trader will attempt to plan his execution trajectory such that he *crosses the spread* infrequently, when there is sufficient quote volume at the *top of the limit order book* and the *spread* is narrow. This would minimise the cost paid for guaranteed execution. Thus, *spread* and *quote volume* were natural candidates for public state attributes if we wish our RL agent to learn this behaviour. In particular, when *spreads* are *narrow (wide)* and *volumes* are *high (low)*, we expect the RL agent to trade *more (less) aggressively* to minimise the trading program's overall execution cost.

While informed by domain knowledge and consistent with the data-preprocessing paradigm described in [2], these choices are somewhat subjective and are only capable of a partial representation of the true state space. Indeed, even the enumeration of all possible *spread* and *volume* configurations at the finest resolution is unlikely to be able to capture the endogenous and exogenous dynamics of the financial system, especially given recent arguments for multi-level causation and scale-specific behaviour ([19,36]). While we can increase the complexity of the state space representation by increasing the number of attributes, the *curse of dimensionality* soon prevents computational tractability for an online algorithm, at least in the *Q-learning* setting we consider.

We propose an alternative notion to characterise the state at each decision point, effectively reducing the set of public attributes to a single metric, while preserving information from *all* measurable aspects of the system.

One can think of a particular realisation of state attributes as a cluster configuration of observable features for a stock. Consider the case of the model used by [21], where *spread* and *quote volume* were used as public attributes. These are derived from the following low-level features of the limit order book: *Level-1 Bid Price, Level-1 Bid Volume, Level-1 Ask Price, Level-1 Ask Volume*. Fig. 1 illustrates how a cluster configuration of these low-level features has an analogous interpretation to the *low/high spread, low/high volume* regimes described in [21].

In time period t_1, we see *Level-1 Ask Volume, Level-1 Bid Volume* and *Level-1 Bid Price* are all correlated and increasing, thus being ascribed to the same cluster. *Level-1 Ask Price* is decreasing and is ascribed to another cluster. In particular, we notice that *Level-1 Bid Price* is *increasing* and *Level-1 Ask Price* is *decreasing*, which is consistent with a *narrowing spread* regime. Since we are considering market orders for a *BUY* trading program, we note that the narrow spread is accompanied by a larger *Level-1 Ask Volume*, which presents favourable conditions for an increase in trading activity. Thus the *low spread, high quote volume* regime considered in the *SSRQ* model has an analogous feature cluster configuration interpretation.

As a further example, consider the cluster configuration in time period t_2. Here, *Level-1 Ask Price* is increasing, while *Level-1 Bid Price* and *Level-1 Ask Volume* are both decreasing, resulting in a *high spread, low quote volume* regime, consistent with a decrease in trading activity.

This simple illustration demonstrates that the cluster configurations of low-level sensory features in high-frequency financial markets may have an analogous interpretation to the trader-intuition-derived regimes usually specified. Furthermore, by allowing the clustering algorithm to be exposed to streaming data from *all* measurable features, unique and persistent cluster configurations may yield meaningful state representations for machine learning classifiers and predictors, beyond those which may have been expected and proposed by human traders. In addition, an appropriate cluster configuration similarity metric can be used to identify temporal states which are characterised by the same feature cluster configurations. If certain configurations persist throughout the trading day, we then have a reduced number of states for which to solve our optimal trading policy. Thus, we will investigate using cluster configurations to describe temporal regimes in a reinforcement learn-