A Computational Social Science Environment for Financial/Economic Experiments

Michal Galas, Dan Brown and Philip Treleaven
University College London, Gower Street, London WC1E 6BT
p.treleaven@ucl.ac.uk

ABSTRACT
This paper describes a major computational social science environment for conducting experiments in finance and economics, specifically for investigating trading algorithms and economic risk.

For the past seven years UCL has worked with the major investment banks developing algorithms for trading systems, and with the regulators investigating high-frequency trading risk and systemic risk. To support this work we have developed two computational environments/platforms: a) our Algorithmic Trading & Risk Analytics Development Environment (ATRADE), and b) our social media streaming, storage and analytics platform (SocialSTORM).

This paper focuses on algorithmic trading (i.e. ATRADE) designed specifically to study the behavior and risk of trading algorithms. In the paper we briefly describe the algorithms used for automated trading and a library of algorithms we have developed which we believe is unique in academia. In addition, ATRADE is used to support an annual global algorithmic trading competition which provides a basis for evaluating performance.

KEYWORDS
Computational environments, big data, finance, economics, algorithmic trading, risk, social media

1. Introduction

Computational Social Science [1] is having a profound impact on social science research, especially in the area of finance and economics research in academia and financial institutions. Two important areas are high-frequency algorithmic trading and risk.

1.1 Algorithmic Trading

In electronic financial markets, algorithmic trading [2, 3, 4] is the use of computer programs to automate one or more stages of the trading process: *pre-trade analysis* (data analysis), *trading signal generation* (buy and sell recommendations), and *trade execution*; furthermore, trade execution is subdivided into two broad categories: agency/broker execution, when a system optimizes the execution of a trade on behalf of a client, and principal/proprietary trading, where an institution is trading on its own account. Each stage of this trading process can be conducted by humans, by humans+algorithms, or fully by algorithms.
When we use the term an ‘algorithmic trading strategy’ we are typically referring to the precise nature of the entire spectrum of algorithms employed by a software system

starting from pre-trade analysis, model selection, signal generation, trade execution and post-trade handling. Strategies usually base their trading on the computed values of some analytical features captured by technical indicators: trend indicators (i.e. momentum), Mean reversion, etc [3]. Three common strategies are:

- **Basic Arbitrage** - arbitrage is the practice of taking advantage of differences in prices of one, or combination of few tradable securities quoted on two or more markets striking a combination of matching deals that capitalize upon the imbalance, and profiting from the difference between the market prices.

- **Momentum** – A trend following trading strategy that aims to capitalize on the continuance of existing trends in the market. The algorithm assumes that large increases in the price of a security will be followed by additional gains and vice versa for declining values.

- **Mean Reversion** – A trading strategy assuming prices and returns eventually move back towards the mean or average. A popular strategy is mean reversion (pairs trading) where two historically correlated securities that have diverged, are assumed to converge in the future.

What makes the investigation of trading algorithms fascinating is that it may involve the optimization of hundreds (or possibly thousands) of variables in a trading system; with even minor changes to a variable having a profound effect on the behavior and more importantly the performance of the system.

1.2 Risk Modeling

Risk modeling research subdivides into what might be termed: a) *portfolio risk* – the risk associated with specific financial instruments (e.g. VaR calculations), and b) *economic risk* – involving endogenous and exogenous risks associated with the economy (e.g. systemic risk). Firms focus on portfolio risk, while government agencies focus on economic risk.

Risk models [2, 3] in algorithmic trading involve the risks associated with a target portfolio of financial instruments and the relevant factors which affect the economic climate and hence the current/future value of the financial instruments. It does so by attempting to quantify both the risk associated with individual instruments and with the portfolio. The two principal approaches to risk which are referred to as:

- **Limiting Amount of Risk** – limiting the size of risk is managing the exposure through a) limiting by constraint or penalty, or b) calculating volatility and dispersion; and

- **Limiting Type of Risk** – this focuses on eliminating whole types of exposure. As with other models the approaches broadly subdivide into: a) Theory-driven and b) Empirical using historical data.

For economic risk, due to major concerns such as the European deficit crisis and the May 6, 2010 Flash Crash, we are investigating high frequency trading risk (e.g. Flash crashes) and systemic risk using agent-based models and even computational ecology models with Government agencies.

2. Algorithmic Trading

Figure 1 illustrates the major algorithm components of a systematic trading system [3, 4], which may be divided into five stages:

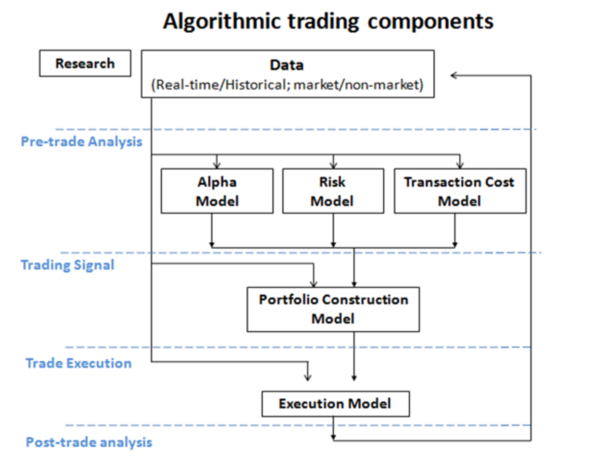

Figure 1: Components of an Algorithmic Trading System

Data access/cleaning – obtaining and cleaning (financial, economic, social) data that will drive algorithmic trading

Pre-trade analysis – analyses properties of assets to identify trading opportunities using market data or financial news etc. (data analysis).

Trading signal generation – identifies the portfolio of assets to be accumulated, based on the pre-trade analysis (what & when to trade).

Trade execution – executing orders for the selected asset (how to trade).

Post-trade analysis - analyses the results of the trading activity, such as P&L, the difference between the price when a buy/sell decision was made and the final execution price (trade analysis).

2.1 Algorithms

The pre-trade analysis comprises three main components: a) *Alpha model* – this is the mathematical model designed to predict the future behavior of the financial instruments that the algorithmic system is intended to trade; b) *Risk model* – evaluates the levels of exposure/risk associated with the financial instruments; and c) *Transaction Cost model* – calculates the (potential) costs associated with trading the financial instruments.

At the Trading Signal stage, the *Portfolio Construction model* takes as its inputs the results of the Alpha model, the Risk model and the Transaction Cost model and decides what portfolio of financial instruments should be owned and in what quantities, in the next time horizon.

Finally at the Trade Execution stage, after the trading signal has been generated and the portfolio constructed, the *Execution Model* performs several decisions with constraints on (actual) transaction costs and trading duration: the most general decision is the Trading Strategy followed by the Venue and Order type, decisions handled by smart order routing. The prerequisites for algorithmic trading include:

- **Centralized Order Book** – shared centralized order book listing the buy and sell orders.

- **Liquid Markets** – a highly liquid market and typically one suitable for high-frequency trading of securities (e.g. equities, bonds, FX).

- **Financial information protocols** – financial information XML exchange protocols (e.g. FIX the Financial Information eXchange protocol [5]) for the computer communication of information.

2.2 Library of Trading Algorithms

Although there are a number of specialist financial system vendors that supply a range of algorithms for automated trading, we are not aware of any academic research group that has developed a library of algorithms that can be used for research.

A natural extension of our research was to develop a library of trading algorithms; simple market-neutral examples of common strategies such as momentum, mean reversion, Bollinger Bands, On-balance volume, ADLine indicator, average directional movement etc. based on a common architecture.

The terms market-neutral and risk-neutral algorithmic trading [3, 6] is defined as:

- **Market-neutral** – market neutral strategies are trading strategies widely used by hedge funds or proprietary traders that go long on certain instruments while shorting others in such a way that the portfolio has no net exposure to broad market moves. The goal is to profit from relative mis-pricings between related instruments - going long on those that are perceived to be underpriced while going short on those that are perceived to be overpriced. Market neutral strategies are sometimes called relative value strategies.

- **Risk-neutral** – are trading strategies insensitive to risk, meaning a strategy which is indifferent to the risk involved in an investment and is only concerned about expected return.

For example, a popular strategy in equity trading is market-neutral, statistical arbitrage (stat arb), pairs trading [3].

2.3 Momentum Algorithm Example

A typical momentum trading strategy for stocks aims at capturing trends in stock prices. It uses the contention that stocks with an upward momentum will continue rising and should be bought and stocks with downward momentum will continue falling and should be sold. We use the nomenclature of "winners" for the former and "losers" for the latter. In our implementation of the momentum strategy, we use the

SMA (simple moving average for the last 60 minutes) – the 60 minute simple moving average metric to capture upward/downward trend.

The strategies commence by conducting *Pre-trade analysis* on stock price data of a specific granularity. Pre-trade analysis usually entails computing the value of one or more technical indicators. For the purpose of the strategy below we have used a single technical indicator SMA (60 minutes). The value of SMA (60 minutes) is used for signal generation.

If the stock price for one of the shortlisted stocks exceeds the value of the SMA (60 minutes) the system generates a buy signal and vice-versa. This stage is usually called *signal generation*. During signal generation the stocks that need to be traded get picked via the signals.

The next phase of the strategy consists of *Portfolio construction*, where we determine the amounts of each stock to hold. A simple portfolio construction rule is to invest equal dollar amounts in each stock that needs to be traded since we do not know how each stock would perform. After we run the strategy for at least a day, we can compute stock specific performance metrics to invest aggressively in specific stocks that performed well.

For the *Trade Execution model* we might assume that there is a fixed strategy update cycle where the strategy does the Pre-trade analysis for the look-back period, scans for trading signals and places trades and this sequence of actions is repeated at regular intervals. Each trade has an investment horizon -the maximum duration one can hold a position (a parameter of the strategy) and we may also temporarily withdraw from trading any stock whose number of incurred consecutive losses or the total unrealized loss exceeds a pre-determined value (another parameter).

2.4 Data Access and Cleaning

In algorithmic trading and risk modeling there is a strong correlation between the quantity/quality of the data available for analysis and the success of the trading system. Data sources broadly comprise:

- **Financial data** – *price* data on financial instruments from Exchanges and electronic communication networks (ECNs), and also financial information from services such as Bloomberg and Thomson Reuters.

- **Economic data** – *fundamental* economic data, such as the overall state of the countries' economies (e.g. unemployment figures), interest rates, gross domestic product and national policy.

- **Social/News data** – *sentiment* data 'scraped' from social media (e.g. TWITTER, Facebook, Google), RSS feeds and news services.

subdividing into:

- **Real-time** – live data feeds from Exchanges, ECNs, news services or steamed social media.

- **Historic** – previously accumulated and stored financial, economic and social/news data.

These data sources, real-time and historic, are the cornerstone of the research, design and back testing of trading algorithms and drive the decision making of all algorithmic trading system components. Buying (raw and especially cleaned) data is hugely expensive and cleaning data is highly time-consuming, but essential due to the sensitivity of trading algorithms.

3. Computational Environments

To conduct large-scale research in finance and economics, researchers need access to terabytes/petabytes of real-time and historic data (e.g. financial, economic, social/news, retail, healthcare, etc.) stored on Hadoop/MapReduce-like data banks [7].

3.1 Financial/Economic Environments

In academia the largest source of 'big' financial data is the Wharton Research Data Services (WRDS) [8] a web-based business data research service from the Wharton School at the University of Pennsylvania. Another (highly ambitious) example of computational social science research is the proposed European FuturICT project [9] a proposal seeking €1b/$1.3b over 10 years to build a 'Living Earth Simulator'; a public domain computational social science and economics facility where major teams of scientists can conduct large-scale data mining and computational modeling on techno-socio-economic-environmental-political systems.

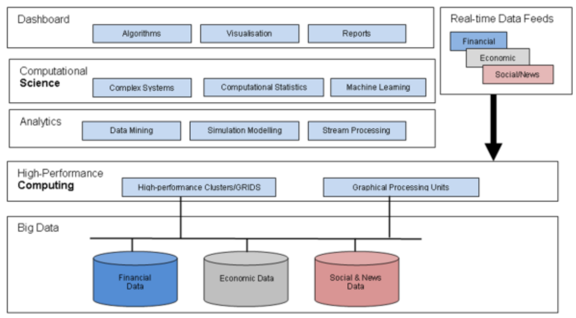

Figure 2: Computational (Social Science) Environment

Figure 2 illustrates the main components of these environments: a) *'big'* databanks; b) *real-time* streamed data feeds such as trading data and social media data; c) *high-performance* computing; d) *analytics* using data mining, simulation modeling and stream processing; e) *computational science* using complex systems, computational statistics and machine learning techniques; and f) a *dashboard* for controlling and visualizing the computations including programmatic control and machine readable reporting.

3.2 UCL's Computational Environments

UCL in collaboration with Microsoft has developed two computational (social science) environments for managing large volumes of real-time and archive data (financial, economic and social media) and supporting data mining, simulation modeling and stream processing analytics using high-performance computer clusters. The specific focus of our environments is algorithmic trading, risk and social sentiment analysis; utilizing real-time financial, social and news data.

The Algorithmic Trading & Risk Analytics Development Environment (ATRADE) platform, which has over 60-person years of programming effort, allows users to conduct experiments virtually or to trade with real funds, with use of a Software Development Kit (SDK) that provides access to a set of programmatic and graphical interfaces for both algorithmic and manual trading. To provide such capability the developers of the platform have designed sets of frameworks, templates and functionalities for AT strategy implementation, including price/order/account/position retrieval, pre/post-trade analytics, trade execution and risk/money management.

The ATRADE platform consists of a set of distributed, multi-threaded, event-driven, real-time, Linux services communicating with each other via an asynchronous messaging system. It provides a proprietary API to support development of algorithmic trading models and strategies. It allows advanced trading-signal generation and analysis in real-time, with use of statistical and technical analysis as well as the data mining methods. It provides data aggregation functionalities to process and store market data feeds. Finally, the platform allows back and forward testing of trading strategies.

The second environment, UCL's social media streaming, storage and analytics platform (SocialSTORM) is a cloud-based 'central hub' platform which facilitates the acquisition of text-based data from online sources such as Twitter, Facebook, RSS media and news. The system includes facilities to upload and run Java-coded simulation models to analyze the aggregated data; which may comprise scraped social data and/or users' own proprietary data. There is also connectivity to the ATRADE platform which supports access to further quantitative finance and economic data, and associated analytics.

SocialSTORM consists of infrastructure tools to facilitate data acquisition, database connectivity, and various levels of access and administration along with data repositories for long and short term data storage. The platform is able to operate in two simulation modes: a 'historical' mode which utilizes data already stored at the time of running the desired simulation (ideal for data-mining and back-testing), and a 'live' mode which operates on a near real-time stream of data which is continually monitored from the sources throughout the simulation (ideal for analyzing financial markets and developing algorithmic trading strategies).

4. Computational Environment and Algorithm Facilities

Given the likely interests of the audience, we will focus on the facilities provided by the environment for algorithm development and testing.

As discussed, the ATRADE and SocialSTORM platforms employ a common system architecture (Figure 3) consisting of a set of distributed, multi-threaded, event-driven, real-time, Linux services communicating with each other via an asynchronous messaging system. For example, the ATRADE platform allows multi-user real and virtual trading. It provides a proprietary API to support development of algorithmic trading models and strategies. It allows an advanced trading-signal generation and analysis in real-time, with use of statistical and technical analysis as well as the data mining methods. It provides data aggregation functionalities to process and store market data feeds. Finally, the platform allows back and forward testing of trading strategies.

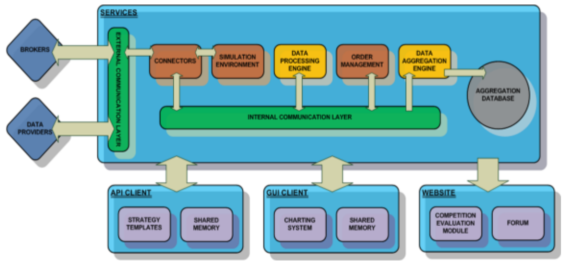

Figure 3: Environment System Architecture and Modules

The common system architecture comprises the following modules: a) *back-end services* - the core of the platform functionalities; b) *Front-End Client APIs* - a set of programmatic and graphical interfaces that can be used to interact with a platform to implement and test analytical models; c) *Connectivity Engine* - provides a means of communication with the outside world; with financial brokers, data providers and others; d) *Internal Communication Layer* - an internal messaging system in the platform draws from the concept of event-driven programming; e) *Aggregation Database* - a fast and robust DBMS functionality, for an entry-level aggregation of data, which is then filtered, enriched, restructured and stored in big data facilities; f) *Client SDK* - a complete set of APIs (Application Programming Interfaces) that enables development, implementation and testing of new analytical models with use of the developers favourite IDE (Integrated Development Environment); g) *Shared Memory* - a buffer-type functionality that speeds up the delivery of temporal/historical data to models and the analytics-related elements of the platform (i.e. the statistical analysis library of methods), and, at the same time, reduces the memory usage requirement; and h) *Model Templates* - the platform supports two generic types of models that ‘push’ and ‘pull’ data streams.

4.1 Algorithm Experimentation Facilities

To support algorithm research, the environment has the following properties.

Simulation & Real Trading. The platform allows algorithms to trade virtually or for real, with use of a dedicated API able to support algorithmic trading. To provide such capability the developers of the platform have designed a set of interfaces that allow issuing and maintenance of orders, extraction and processing of market data to generate trading signals, extraction and processing of order data and finally management of information related to P&L and users’ account information.

Rapid Prototyping. The platform is a framework for developing, trading, testing and evaluating the ‘algorithm’ risk. Rapid prototyping requires usage of a simple programming language and also simple and powerful interfaces that provide as much embedded functionality as possible without a need of a heavy, low-level programming. ATRADE meets all the requirements.

Data Processing & Aggregation. The platform is capable of aggregating and processing data in real-time. Information is a key to success in algorithmic trading, therefore, ATRADE aggregates and processes data in real-time and delivers it to the users for analysis. Apart from delivering the data in real-time, the platform allows retrieval of historical data through a set of dedicated interfaces.

R & Matlab. The platform provides capability of incorporating statistical/mathematical computing environments. Statistical and mathematical computing concepts are particularly applicable to modeling and implementation of algorithmic trading strategies.

Black Box and Multiple Models. A particularly useful functionality of the platform is a module for automated evaluation of risk of black box models that allows ranking of the models on such basis; without a need of handing the source of the models to anyone. This feature is useful in eliminating the IPR issues and ensures safety of the algorithmic trading models from a potential third party threats.

5. Experimental Evaluation

The performance of the Computational Environment has been thoroughly tested by the UCL global Algorithmic Trading Competition held annually for students, researchers and academics to test their trading algorithms. Each year the competition allows competitors to research and design their algorithmic solutions, get experience with trading by implementing their algorithms, and finally compete for prizes of up to $40,000. It also allows UCL to test and evaluate the capabilities of the Environment.

The biggest challenges in enterprise-size financial systems are performance, scaling (with respect to the amount of data, amount of users and amount of trading algorithms being executed concurrently), variety of types of latencies and fault tolerance of the system. Different elements of the environment as well as the entire system are tested periodically during development, refactoring and during the live events organized on the basis of the platform.

In 2010 the platform was run with 100 HFT models running in parallel over a test period of one week. The models were designed to be able to subscribe to market events of 20 major currency pairs and to issue order instructions each time a new market event occurred; it is common that at least a few events occur each second, for every currency pair. During the 2011 competition over 300 traders used the platform to trade with one or more models. Over the last two years the platform has been tested with usage between 10 to 70 streams of live market feeds of different liquid securities; it exhibited no significant issues with internal performance or latency.

Table 1 lists statistics on tick data from Trading Technologies (TT) and Knight Capital (HotSpotFX), and shows the current amount of inbound tick data the ATRADE platform is able to handle:

TT Connector	HotSpotFX Connector	
300	265	The average message size (Bytes)
20	60	Amount of supported streams
2	7	The average amount of messages per stream (second)
12	111.3	The average bandwidth usage (KBytes/second)

Table 1: Inbound Tick Data

Table 2 lists statistics relate to the current level of users and strategies managed by the ATRADE platform:

Mean	Deviation	Amount of Trials	
230	52	100	Amount of concurrent instances of SDK (one per user), the ATRADE platform can manage without significant performance degradation
579	67	100	Amount of concurrent instances of Strategies (per single SDK), the ATRADE platform can handle without becoming unstable
160:75	24:48	100	An average amount of instances of SDK and Strategies the ATRADE platform can manage without significant performance degradation

Table 2: User Statistics

For test evaluation, the above sets of experiments were performed with an individual (non-scaled) instance of ATRADE services, hosted on a single Dell Power Edge R300 Server. All the instances of the ATRADE SDK, that were handling the running strategies, were hosted on a single ACER ASPIRE 5935G Laptop.

The above results emphasize the worst-case scenario performance, *at an unscaled level*, where the software is limited by the underlying performance of a single hardware unit. Thanks to the modularization of the platform and utilization of highly scalable technologies, the results scale linearly with additional server and client machines.

Over the course of the 2011 Algorithmic Trading Competition some problems were observed with error detection and correction, especially during the peak hours. A diagnostics and monitoring service was deployed to evaluate, monitor and report on the work of ATRADE services in runtime. The service is capable of restarting elements of ATRADE that exhibits problems and is able to e-mail reports on such events to the administrator of the system.

6. ATRADE and Traders

Competitions organized with ATRADE provide unique chance to observe and evaluate behavior of different models. Researchers may be interested in different trading approaches, main behavioral drives in models and motivation behind model actions. They key challenge in this respect is related to a lack of access to source code of different models that prevents researchers from direct evaluation. ATRADE's approach to work around this problem is based on a black box assessment; where, rather than looking directly at the strategies, the system is capable of evaluating model's behavior on the basis of its footprint left in the system. Such footprint is composed of order-instructions issues to the system by each model in the trading process that affects risk and performance metrics of traders' holdings.

To enable support for competition evaluation, one of ATRADE modules is designed to periodically (usually on a daily basis) sample various risk and performance metrics generated for each participating trader. Such time-series of samples allowed assessment of changes in the traders' holding and thus, on the basis of rates of change in the metrics we were able to infer some information about models.

6.1 Diversity of approaches across traders

Algorithmic traders exhibit a diverse set of approaches to trading. The most common is intraday approach with a few dozen of transactions being executed to adjust and maintain positions through the day. Such positions are then being closed at the end of every day. In this approach traders usually look for daily trends and news-announcements they can benefit from. Another popular approach is the high-frequency approach, where traders attempt to benefit from extremely short market inefficiencies with bursts of high frequency transactions. In this approach the traders look for inefficiencies in pricing, volumes and liquidity and the positions are not being held for longer than few seconds. Finally, a third popular approach across traders is a long-term portfolio asset allocation, where traders maintain positions on the basis of long-term trends, social sentiments and fundamental economic prognosis.

6.2 Behavior of traders and trading models

One may attempt to identify typical stimulus for traders and their models as well as to infer a typical set of actions such strategies exhibit with respect to stimulus. Based on the data collected during the competitions we can summarize our observations as follows:

- **Stimulus based on time, traded volumes and available liquidity** – timing, margin, traded volumes and available liquidity can be considered the most basic stimulus influencing behavior of every trader and consequently being reflected in their trading strategies.
- **Stimulus based on PnL, risk and performance information** – a feedback stimulus on the basis of already held positions in traded securities. Traders may apply performance and risk measures to their positions and perform analysis on the types of actions that can be beneficial at a given threshold of a measure that is being taken into account.
- **Stimulus based on fundamental analysis** – traders may perform fundamental analysis of a state of health of particular security and may evaluate whether the security is undervalued or overvalued. On that basis they may create a strategy that will attempt to exploit the security fundamentals. The typical behavior of a trader will involve fundamental analysis and selection of the most promising securities on such basis.
- **Stimulus based on technical/quantitative analysis** – traders may perform statistical analysis of historical data on a set of chosen technical indicators or quantitative measures to attempt to spot exploitable patterns in the data. On the basis of such patterns the traders may then attempt to create a strategy.
- **Stimulus based on news and social analysis** – traders may perform analysis of news announcements or sentiment analysis of social media to try to find exploitable patterns with relation to which they can trade. This can be then incorporated into a strategy.

6.3 Motivation of traders

One may be interested in attempting to understand why traders and consequently their models act in a certain way and what motivates their actions. Arguably maximization of profit and minimization of risk may be considered the strongest drives in most circumstances. Most ‘quantitative’ traders generalized this idea with utility theory and construct their models to exhibit optimal behavior with respect to appropriately weighted multi-factor utility function. Apart from previously mentioned measures of performance and risk, the main factors in such function ma by for example a drive towards stable growth of return or an appetite for hedging away the risk.

7. Conclusions

It is important that social science researchers have access to computational environments and ‘big’ data for experimentation. Otherwise computational social science could become the exclusive domain of major companies, government agencies and a privileged set of academic researchers presiding over private data from which they produce papers that cannot be critiqued or replicated. Arguably what is required are public-domain computational environments and ‘big’ data facilities for quantitative social science (e.g. economics), that can be accessed by researchers via a cloud-based facility.

Our contribution to these computational environments is the platforms and common system architecture developed financial and economic research.

8. References

[1] Lazer, D., et al., “Computational Social Science”, Science, 6 February 2009, Vol. 323 no. 5915 pp. 721-723.

[2] Chan, E. P., Quantitative Trading: How to Build Your Own Algorithmic Trading Business (Wiley Trading), 2009.

[3] Narang, R. K., Inside the Black Box: The Simple Truth About Quantitative Trading (Wiley Finance), 2009.

[4] Nuti, G., Mirghaemi, M., Treleaven, P., and Yingsaeree, C., Algorithmic Trading, IEEE COMPUTER, Volume: 44 Issue: 11, page(s): 61 – 69, Nov. 2011.

[5] Financial Information eXchange, Wikipedia, the free encyclopedia. 2010. http://en.wikipedia.org/wiki/Financial_Information_eXchange.

[6] Stokes, E., Market Neutral Investing, Dearborn Trade Publishing, 2004.

[7] Abouzeid, A., Pawlikowski, Abadi, K., Rasin, A., and Silberschatz, A. "HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads", PVLDB,2(1):922–933, 2009.

[8] Wharton Research Data Services (WRDS), https://wrds-web.wharton.upenn.edu/wrds/

[9] FuturICT Project, www.futurict.eu

Overview, Design concepts and Details (ODD) Protocol

We have done our best to address the ODD protocol. This paper describes a computational social science environment (rather than a specific model) for conducting experiments in finance and economics, specifically for investigating trading algorithms and economic risk.

Overview

This paper focuses on algorithmic trading (i.e. ATRADE environment) which is used for experimentation with virtual and real trading, and has been designed specifically to study the behavior and risk of trading algorithms. In the paper we briefly describe the algorithms used for automated trading and a library of algorithms we have developed which we believe is unique in academia. In addition, ATRADE is used to support an annual global algorithmic trading competition which provides a basis for evaluating performance.

Purpose

- The focus is a computational social science environment for investigating trading algorithms and economic risk.

State Variables and Scales

- As discussed, we describe the structure of algorithms used for trading and out computational environment, and we as an example present an overview of a momentum based strategy. Data access and cleaning is also discussed.

Process Overview and Scheduling

- The operation of an algorithmic trading system and the operation of our computational social science environment is described in detail.

Design Concepts

Design Concepts

- Design of an algorithmic trading system with a momentum based algorithm strategy is presented in significant detail.

Details

- Details of algorithmic trading and the environment are presented. Although this is clearly difficult due to the nature of the paper.

Initialisation

- A section is included on the sources of data and on strategies for data cleaning.

Input

- This is covered by the data access section.

Submodels

- The structure of algorithmic trading systems is described.