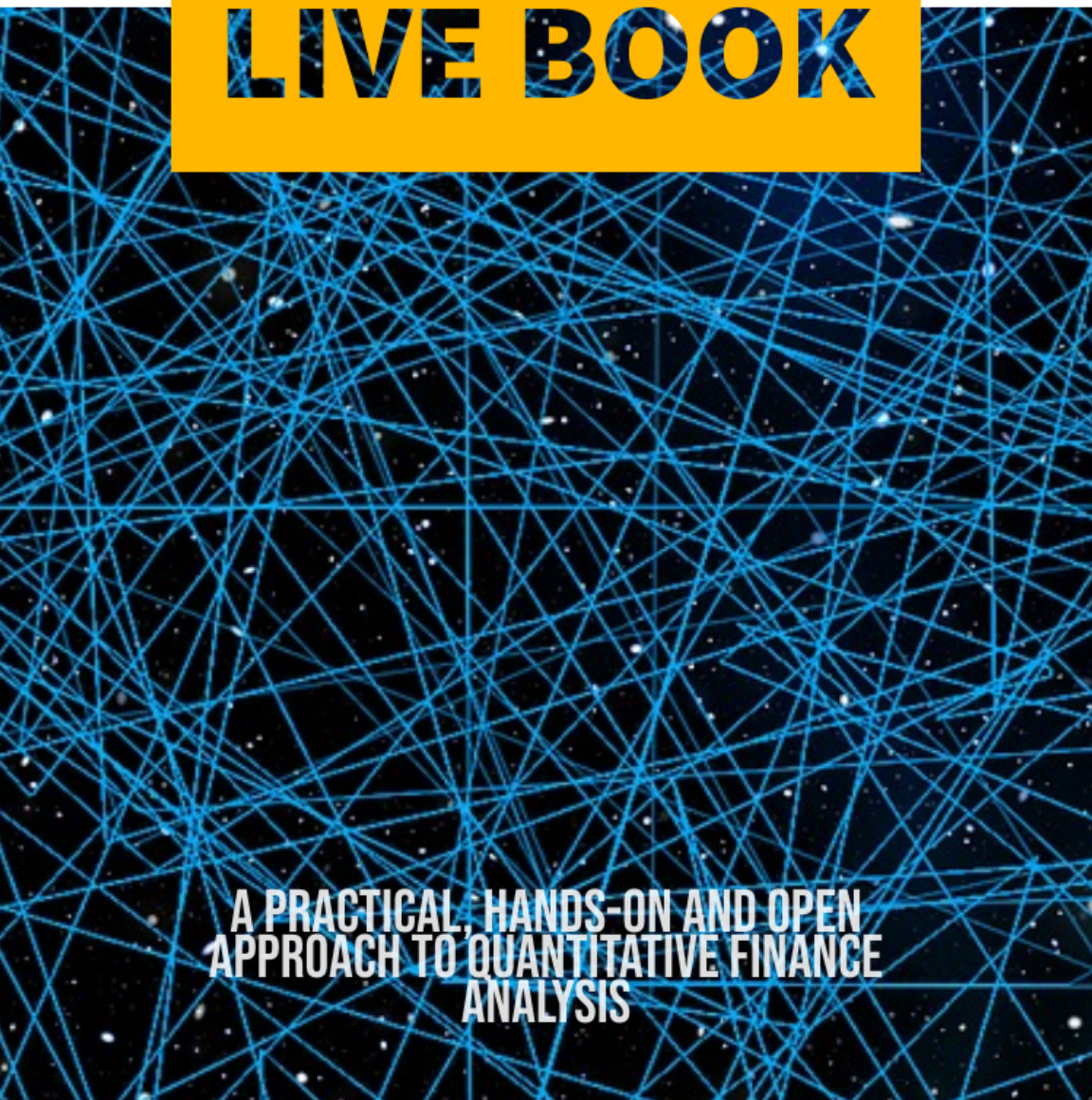
Thársis T. P. Souza

OPEN QUANT LIVE BOOK



A PRACTICAL, HANDS-ON AND OPEN APPROACH TO QUANTITATIVE FINANCE ANALYSIS

The Open Quant Live Book

Thársis T. P. Souza

2019-01-06

Contents

Preface

Working Contents

1. The Basics

- I/O
- Stylized Facts
- Correlation & Causation

2. Algo Trading

- Investment Process
- Backtesting
- Factor Investing
- Limit Order

3. Portfolio Optimization

- Modern Portfolio Theory
- Measuring Risk
- Linear Programming

4. Machine Learning

- Intro
- Agent-Based Models
- Binary Classifiers
- AutoML
- Hierarchical Risk Parity

5. Econophysics

- Entropy, Efficiency and Coupling
- Transfer Entropy, Information Transfer and Causality
- Financial Networks

6. Alternative Data

Contribute

The Book is Open[1] and we are looking for co-authors. Feel free to reach out[2] or simply create a pull request with your contribution on our Github project[3].

Book's information

First published at: openquant.netlify.com[4].

[1] https://github.com/souzatharsis/open-quant-live-book
[2] http://www.souzatharsis.com/
[3] https://github.com/souzatharsis/open-quant-live-book
[4] https://openquant.netlify.com/
[5] https://creativecommons.org/licenses/by-nc-sa/4.0/

Part I

The Basics

Chapter 1

I/O

In this Chapter, we will introduce basic functions to read text and excel files as well as large files. We will also show how to obtain free financial and economic data from sources such as Quandl, IEX and Alpha Vantage.

1.1 Reading and Writing

1.1.1 Text Files

The most basic and commonly used option to import data from text files in R is the use of the function `read.table` from the **r-base**. We can use this function to read text files with extensions such as `.txt` and `.csv`.

```
dat.table <- read.table(file = "<name of your file>.txt")
dat.csv <- read.csv(file = "<name of your file>.csv")
```

The package **readr** provides functions for reading text data into R that are much faster that the functions from the **r-base**. The `read_table`

function from the package **readr** provides a near-replacement for the `read.table` function.

```
library(readr)
dat.table <- readr::read_table2(file = "<name of your file>.txt")
dat.csv <- readr::read_csv(file = "<name of your file>.csv")
```

Another option to save data is to write it in `rds` format. Data stored in `rds` format has the advantage to keep the original data struture and type of the object saved. Also, `.rds` files are compressed and consume less space than files saved in `.csv` format. A data.frame object can be saved in `rds` format and then loaded back as follows:

```
write_rds(dat.frame, path = "<name of your file>.rds")
dat.frame <- read_rds(path = "<name of your file>.rds")
```

1.1.2 Excel Files

The package `readxl` has an ease to use interface to functions that load excel documents in R. The functions `read_xls` and `read_xlsx` can be used to read excel files as follows:

```
library(readxl)
readxl::read_xls(path = "<name of your file>.xls")
readxl::read_xlsx(path = "<name of your file>.xlsx")
```

The function `read_excel()` automatically detects the extension of the input file as follows:

```
readxl::read_excel("<name and extension of your file>", sheet = "<sh
```

In the `read_excel` function, the `sheet` argument can receive either the target sheet name or index number, where sheet indexing starts at 1.

The `readxl` has been oberving increased use compared to other comparable packages such as **gdata** and the **xlsx** due to its relative ease of use and performance. Also, the `readxl` do not have depency with external code libraries while the packages **gdata** and **xlsx** depend on `ActiveState PERL` and the `Java JDK`, respectively.

1.1.3 Large Files

Fast data manipulation in a short and flexible syntax.

1.2 Data Sources

In this section, we will show how to obtain financial and economic data from public sources.

1.2.1 Alpha Vantage

Alpha Vantage offers free access to pricing data including:

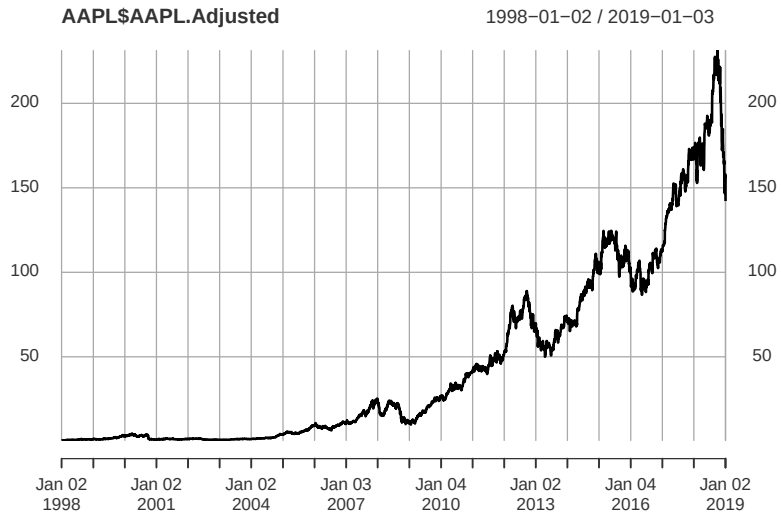
- Stock Time Series Data;
- Physical and Digital/Crypto Currencies (e.g., Bitcoin);
- Technical Indicators and
- Sector Performances.

The data are available in JSON and CSV formats via REST APIs. The **quantmod** and the **alphavantager** R packages offer a lightweight R interface to the Alpha Vantage API. Daily stock prices can be obtained with the `quantmod::getSymbols` function as follows:

```
getSymbols(Symbols = "AAPL", src = "av", output.size = "full",
  adjusted = TRUE, api.key = "your API key")
```

The output data is stored in an object with the same name as the corresponding symbol, in this example `AAPL`. The output data looks like the following

AAPL.Open	AAPL.High	AAPL.Low	AAPL.Close	AAPL.Volume	AA
13.6	16.2	13.5	16.2	6411700	
16.5	16.6	15.2	15.9	5820300	
15.9	20.0	14.8	18.9	16182800	
18.8	19.0	17.3	17.5	9300200	
17.4	18.6	16.9	18.2	6910900	
18.1	19.4	17.5	18.2	7915600	



We called the `quantmod::getSymbols` function with the following arguments:

- `Symbols='AAPL'` defines a character vector specifying the names of each symbol to be loaded, here specified by the symbol of the company Apple Inc.;
- `src="av"` specifies the sourcing method, here defined with the value corresponding to Alpha Vantage;
- `output.size="full"`specified length of the time series returned. The strings `compact` and `full` are accepted with the following specifications: `compact` returns only the latest 100 data points;

`full` returns the full-length time series of up to 20 years of historical data;
- `adjusted=TRUE` defines a boolean variable to include a column of closing prices adjusted for dividends and splits;
- `api.key` specifies your Alpha Vantage API key.

1.2.2 IEX

The IEX Group operates the Investors Exchange (IEX), a stock exchange for U.S. equities that is built for investors and companies. IEX offers U.S. reference and market data including end-of-day and intraday pricing data. IEX offers an API with "a set of services designed for developers and engineers. It can be used to build high-quality apps and services". Data sourced from the IEX API is freely available for commercial subject to conditions[1] and the use of their API is subject to additional terms of use[2].

IEX lists the following github project as an unofficial API for R: https://github.com/imanuelcostigan/iex. We will provide examples on how to obtain intraday pricing data using this package. First, we will use the **devtools** to install the package directly from its github repository as follows:

```
library(devtools)
install_github("imanuelcostigan/iex")
```

The **iex** package provides 4 set of functions as follows:

- `last`: Provides IEX near real time last sale price, size and time. Last is ideal for developers that need a lightweight stock quote. IEX API real time API documentation[3].
- `market`: Provides exchange trade volume data in near real time. IEX market API documentation[4].

[1] https://iextrading.com/api-exhibit-a/
[2] https://iextrading.com/api-terms/
[3] https://iextrading.com/developer/docs/#last
[4] https://iextrading.com/developer/#market-market

- `stats`: A set of functions that return trading statistics. IEX stats API documentation[5].
- `tops`: Provides IEX's aggregated bid and offer position in near real time for all securities on IEX's displayed limit order book. IEX API TOPS documentation[6].

For instance, the `last` function has the following arguments:

- `symbols`: A vector of tickers (case insensitive). Special characters will be escaped. A list of eligible symbols is published daily[7] by the IEX. When set to `NULL` (default) returns values for all symbols.
- `fields`: A vector of fields names to return (case sensitive). When set to `NULL` (default) returns values for all fields.
- `version`: The API version number, which is used to define the API URL.

We can obtain intraday stock price data with the `last` function as follows:

```
dat <- iex::last(symbols = c("AAPL"), fields = c("symbol",
  "price", "size"))
```

The function returns an S3 object of class `iex_api` which has three accessible fields: `path` , `response` and `content`.

- The `path` contains the corresponding IEX API path:

```
dat$path
```

```
## [1] "tops/last"
```

- The `response` contains the unparsed IEX API response:

```
dat$response
```

```
## Response [https://api.iextrading.com/1.0/tops/last?symbols=AAPL&f
##   Date: 2019-01-04 06:53
##   Status: 200
```

[5] https://iextrading.com/developer/#stats
[6] https://iextrading.com/developer/#tops-tops
[7] https://iextrading.com/trading/eligible-symbols/

```
##   Content-Type: application/json; charset=utf-8
##   Size: 45 B
```

- The `content` contains the parsed content from the API's response:

```
dat$content
```

```
## [[1]]
## [[1]]$symbol
## [1] "AAPL"
##
## [[1]]$price
## [1] 142
##
## [[1]]$size
## [1] 100
```

According to the developer, this package causes R to pause 0.2 seconds after executing an API call to avoid the user being throttled by the IEX API (which enforces a 5 request per second limit). Documentation about the other set of functions can be obtained at https://github.com/imanuelcostigan/iex/tree/master/man.

1.2.3 Quandl

1.3 Conclusion

- We showed how to load and import data from both local files and external sources.
- We provided examples on how to read tabular data and how to handle large files.
- We showed how to obtain financial and economic data from freely available sources.

1.3.1 Key Packages Used

- Importing Flat Files ++ **readr** and **data.table**
- Obtaining Financial Data ++ **quantmod**, **imanuelcostigan/iex** and **quandl**

1.3.2 Further Reading

We recommend the book R for Data Science: Import, Tidy, Transform, Visualize, and Model Data. Additional relevant R packages include:

- dplyr: Fast data frames manipulation and database query.
- reshape2: Flexibly rearrange, reshape and aggregate data.
- readr: A fast and friendly way to read tabular data into R.
- tidyr: Easily tidy data with spread and gather functions.
- rlist: A toolbox for non-tabular data manipulation with lists.
- jsonlite: A robust and quick way to parse JSON files in R.
- ff: Data structures designed to store large datasets.
- lubridate: A set of functions to work with dates and times.

Chapter 2

Stylized Facts

2.1 Introduction

2.2 Distribution of Returns

2.2.1 Fat Tails

A distribuição de retornos financeiros apresenta leptokurtose. A ocorrência de eventos extremos é mais provável comparado com uma distribuição normal, i.e., as caudas da distribuição empírica de retornos são mais "pesadas" comparadas com as caudas esperadas supondo uma distribuição normal de probabilidade.

2.2.2 Skewness

A distribuição empírica de retornos é distorcida para esquerda. Retornos negativos são mais prováveis que retornos positivos.

2.3 Volatility

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2} \tag{2.1}$$

2.3.1 Time-invariance

A volatilidade de retornos financeiros não é constante ao longo do tempo.

2.3.2 Volatility Clustering

Eventos extremos são observados próximos um dos outros.

2.3.3 Correlation with Trading Volume

O volume de negociação de um ativo tem correlação significante com a volatilidade do mesmo.

2.4 Correlation

$$\rho = \frac{\sum_{t=1}^{T}(r_t - \hat{r}_t)(s_t - \hat{s}_t)}{\sqrt{\sum_{t=1}^{T}(r_t^\tau - \hat{r}_t^\tau)^2}\sqrt{\sum_{t=1}^{T}(s_t - \hat{s}_t)^2}}, \tag{2.2}$$

onde $\hat{r}_t$ e $\hat{s}_t$ são a média amostral de r_t e s_t, respectivamente.

2.4.1 Time-invariance

A correlação entre duas series temporais de retornos financeiros não é constante ao longo do tempo.

2.4.2 Auto-correlation

Retornos financeiros apresentam baixa autocorrelação (linear), exceto em escalas de tempo muito baixas, e.g., minutos, onde há presença de efeitos de microstructura. Por outro lado, a função de autocorrelação do valor absoluto de retornos financeiros decai lentamente com o tempo.

A correlação contemporânea é maior do que a correlação cruzada.

Chapter 3

Correlation & Causation

3.1 Introduction

3.2 A First Definition of Causality

We quantify causality by using the notion of the causal relation introduced by Granger where a signal X is said to Granger-cause Y if the future realizations of Y can be better explained using the past information from X and Y rather than Y alone.

The most common definitions of Granger-causality rely on the prediction of a future value of the variable Y by using the past values of X and Y itself. In that form, X is said to G-cause Y if the use of X improves the prediction of Y.

Let X_t be a random variable associated at time t while X^t represents the collection of random variables up to time t. We consider X_t, Y_t and Z_t to be three stochastic processes. Let $\hat{Y}_{t+1}$ be a predictor for the value of the variable Y at time $t+1$.

We compare the expected value of a loss function $g(e)$ with the error $e = \hat{Y}_{t+1} - Y_{t+1}$ of two models:

1. The expected value of the prediction error given only Y^t

$$\mathcal{R}(Y^{t+1} \mid Y^t, Z^t) = \mathbb{E}[g(Y_{t+1} - f_1(X^t, Z^t))] \tag{3.1}$$

2. The expected value of the prediction error given Y^t and X^t

$$\mathcal{R}(Y^{t+1} \mid X^t, Y^t, Z^t) = \mathbb{E}[g(Y_{t+1} - f_2(X^t, Y^t, Z^t))]. \tag{3.2}$$

In both models, the functions $f_1(.)$ and $f_2(.)$ are chosen to minimize the expected value of the loss function. In most cases, these functions are retrieved with linear and, possibly, with nonlinear regressions. Typical forms for $g(.)$ are the $l1$- or $l2$-norms.

We can now provide our first definition of statistical causality under the Granger causal notion as follows:
Definition 3.1. X does not Granger-cause Y relative to side information Z if and only if $\mathcal{R}(Y_{t+1} \mid X^t, Y^t, Z^t) = \mathcal{R}(Y_{t+1} \mid Y^t, Z^t)$.

A more general definition than @ref{def:G1} that does not depend on assuming prediction functions can be formulated by considering conditional probabilities. A probabilistic definition of G-causality assumes that Y_{t+1} and X^t are independent given the past information (X^t, Y^t) if and only if $p(Y_{t+1} \mid X^t, Y^t, Z^t) = p(Y_{t+1} \mid Y^t, Z^t)$, where $p(. \mid .)$ represents the conditional probability distribution. In other words, omitting past information from X does not change the probability distribution of Y. This leads to our second definition of statistical causality as follows:
Definition 3.2. X does not Granger-cause Y relative to side information Z if and only if $Y_{t+1} \perp\!\!\!\perp X^t \mid Y^t, Z^t$.

Def. @ref{def:G2} does not assume any functional form in the coupling between X and Y. Nevertheless, it requires a method to assess their conditional dependency.

In the next Section, we define a parametric linear specification of G-causality based on Def. 3.1.

3.3 Quantifying Granger-Causality

3.3.1 Model Specification

Standard Granger-causality tests assume a linear relationship among the causes and effects and are implemented by fitting autoregressive models **??**.

Consider the linear vector-autoregressive (VAR) equations:

$$Y(t) = \alpha + \sum_{\Delta t=1}^{k} \beta_{\Delta t} Y(t - \Delta t) + \epsilon_t, \tag{3.3}$$

$$Y(t) = \hat{\alpha} + \sum_{\Delta t=1}^{k} \hat{\beta}_{\Delta t} Y(t - \Delta t) + \sum_{\Delta t=1}^{k} \hat{\gamma}_{\Delta t} X(t - \Delta t) + \hat{\epsilon}_t, \tag{3.4}$$

where k is the number of lags considered.

From Def 3.1, X does not G-cause Y if and only if the prediction errors of X in the restricted Eq. (3.3) and unrestricted regression models Eq. (3.4) are equal (i.e., they are statistically indistinguishable).

3.3.2 Test for Misspecification

A statistically significant causality can be reported only if the linear models from Eqs. (3.3) and (3.4) are not misspecified. For that purpose, we utilize the BDS test **?** for the model misspecification (see Section **??**).

The BDS test **?** is used to detect nonlinear dependence in time series. When applied to the residuals of a linear model, the BDS tests the null hypothesis that these residuals are independent and identically distributed. The BDS test is a powerful test to detect linear misspecification and nonlinearity **??**.

Let $\epsilon_t = (\epsilon_{t=1}, \dots, \epsilon_{t=n})$ be the residuals of the linear fitted model and define its m-embedding as $\epsilon_t^m = (\epsilon_t, \epsilon_{t-1}, \dots, \epsilon_{t-m+1})$. The m-embedding correlation integral is given by

$$C_{m,n}(\Delta\epsilon) = \frac{2}{k(k-1)} \sum_{s=1}^{t} \sum_{t=s}^{n} \chi(\|\epsilon_s^m - \epsilon_t^m\|, \Delta\epsilon),$$

and

$$C_m(\Delta\epsilon) = \lim_{n\to\infty} C_{m,n}(\Delta\epsilon),$$

where χ is an indicator function where $\chi(\|\epsilon_s^m - \epsilon_t^m\|, \Delta\epsilon) = 1$ if $\|\epsilon_s^m - \epsilon_t^m\| < \Delta\epsilon$ and zero, otherwise.

The null hypothesis of the BDS test assumes that ϵ_t is iid. In this case,

$$C_m(\Delta\epsilon) = C_1(\Delta\epsilon)^m.$$

The BDS statistic is a measure of the extent that this relation holds in the data. This statistic is given by the following:

$$V_m(\Delta\epsilon) = \sqrt{n}\frac{C_m(\Delta\epsilon) - C_1(\Delta\epsilon)^m}{\sigma_m(\Delta\epsilon)},$$

where $\sigma_m(\Delta\epsilon)$ can be estimated as described in **?**.

The null hypothesis of the BDS test indicates that the model tested is not misspecified and it is rejected at the 5% significance level if $\|V_m(\Delta\epsilon)\| > 1.96$.

The parameter $\Delta\epsilon$ is commonly set as a factor of the variance (σ_ϵ) of ϵ. We report results for $\Delta\epsilon = \sigma_\epsilon/2$ and the embedding dimension $m = 2$. We also performed tests for $\Delta\epsilon = \sigma_\epsilon$ and $m = 3$ with no significant differences in the results.

3.3.3 Analysis of Variance

A one-way ANOVA test is utilized to test if the residuals from Eqs. (3.3) and (3.4) differ from each other significantly.

3.3.4 Multiple Hypotheses Testing Correction

When more than one lag k is tested, a Bonferroni correction is applied to control for multiple hypotheses testing.

Part II

Algo Trading

Chapter 4

Limit Order

Part III

Portfolio Optimization

Part IV

Machine Learning

Part V

Econophysics

Chapter 5

Entropy

Let X be a random variable and $P_X(x)$ be its probability density function (pdf). The entropy $H(X)$ is a measure of the uncertainty of X and is defined in the discrete case as follows:

$$H(X) = -\sum_{x \in X} P_X(x) \log P_X(x). \tag{5.1}$$

If the log is taken to base two, then the unit of H is the *bit* (binary digit). We employ the natural logarithm which implies the unit in *nat* (natural unit of information).

Given a coupled system (X, Y), where $P_Y(y)$ is the pdf of the random variable Y and $P_{X,Y}$ is the joint pdf between X and Y, the joint entropy between X and Y is given by the following:

$$H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} P_{X,Y}(x, y) \log P_{X,Y}(x, y). \tag{5.2}$$

The conditional entropy is defined by the following:

$$H\left(Y|X\right) = H(X, Y) - H(X). \tag{5.3}$$

We can interpret $H(Y|X)$ as the uncertainty of Y given a realization of X.

5.1 Market Efficiency and Bubbles

5.2 Quantifying Non-linear Correlation

Chapter 6

Transfer Entropy

6.1 Introduction

6.2 Nonlinear G-Causality

To compute the nonlinear G-Causality, we use the concept of Transfer Entropy that, since its introduction by Schreiber (2000) **?**, has been recognized as an important tool in the analysis of causal relationships in nonlinear systems **?**. %It detects directional and dynamical information **?** while not assuming any particular functional form to describe interactions among systems.

The Transfer Entropy can be defined as the difference between the conditional entropies:

$$TE\left(X \to Y|\, Z\right) = H\left(Y^F \middle| Y^P, Z^P\right) - H\left(Y^F \middle| X^P, Y^P, Z^P\right), \quad (6.1)$$

which can be rewritten as a sum of Shannon entropies:

$$TE\left(X \to Y\right) = H\left(Y^P, X^P\right) - H\left(Y^F, Y^P, X^P\right) + H\left(Y^F, Y^P\right) - H\left(Y^P\right), \tag{6.2}$$

where Y^F is a forward time-shifted version of Y at lag Δt relatively to the past time-series X^P, Y^P and Z^P. Within this framework we say that X does not G-cause Y relative to side information Z if and only if $H\left(Y^F \middle| Y^P, Z^P\right) = H\left(Y^F \middle| X^P, Y^P, Z^P\right)$, i.e., when $TE\left(X \to Y, Z^P\right) = 0$.

Empirically, we reject this null hypothesis of causality if the Transfer Entropy from X to Y is significantly higher than the shuffled version of the original data.

For this we estimate 400 replicates of $TE(X_{Shuffled} \to Y)$, where $X_{Shuffled}$ is a random permutation of X relatively to Y. We compute the randomized Transfer Entropy at each permutation for each time-shift (Δt) from 1 to 10 days. We then calculated the frequency at which the observed Transfer Entropy was equal or more extreme than the randomized Transfer Entropy. The statistical significance was assessed using p-value < 0.05 after Bonferroni correction.

6.3 The Link Between Linear Granger-causality and Transfer Entropy

Barnett et al. (2009) **?** showed that linear G-causality and Transfer Entropy are equivalent if all processes (X and Y) are jointly Gaussian. In particularly, by assuming the standard measure ($l2$-norm loss function) of linear G-causality for the bivariate case as

$$GC_{X \to Y} = \log\left(\frac{var(\epsilon_t)}{var(\hat{\epsilon}_t)}\right), \tag{6.3}$$

the following can be proved **?**:

$$TE_{X\to Y} = GC_{X\to Y}/2. \tag{6.4}$$

This result provides a direct mapping between the Transfer Entropy and the linear G-causality implemented in the standard VAR framework. Hence, it is possible to estimate the TE both in its general form and with its equivalent form for linear G-causality.

6.4 Net Information Flow

Transfer-entropy is an asymmetric measure, i.e., $T_{X\to Y} \neq T_{Y\to X}$, and it thus allows the quantification of the directional coupling between systems. The Net Information Flow is defined as

$$\widehat{TE}_{X\to Y} = TE_{X\to Y} - TE_{Y\to X}\ . \tag{6.5}$$

One can interpret this quantity as a measure of the dominant direction of the information flow. In other words, a positive result indicates a dominant information flow from X to Y compared to the other direction or, similarly, it indicates which system provides more predictive information about the other system **?**.

For the nonlinear case, the Transfer Entropy was computed as defined in Eq. (6.1). Conversely, to estimate the linear version of the Net Information Flow, we computed the Transfer Entropy using Eq (6.4), i.e., we estimated the linear G-causality (6.3) and multiplied it by a factor of 1/2.

In the next section, we construct simulated systems and test the nonlinear and linear formulations of the net information flow. We show that only the nonlinear formulation of net information flow is able to capture the nonlinear relationships in the simulated systems.

6.5 Empirical Experiment: Information Flow on Simulated Systems

6.6 Empirical Experiment: Information Flow on Global Markets

Chapter 7

Financial Networks

7.1 Introduction

Financial markets can be regarded as a complex network in which nodes represent different financial assets and edges represent one or many types of relationships among those assets. Filtered correlation-based networks have successfully been used in the literature to study financial markets structure particularly from observational data derived from empirical financial time series **??????**. The underlying principle is to use correlations from empirical financial time series to construct a sparse network representing the most relevant connections. Analyses on filtered correlation-based networks for information extraction **??** have widely been used to explain market interconnectedness from high-dimensional data. Applications include asset allocation **??**, market stability assessments **?**, hierarchical structure analyses **?????** and the identification of lead-lag relationships **?**.

7.2 Network Construction

We selected $N = 100$ of the most capitalized companies that were part of the S&P500 index from 09/05/2012 to 08/25/2017. The list of these companies' ticker symbols is reported in the Appendix **??**. For each stock i the financial variable was defined as the daily stock's log-return $R_i(\tau)$ at time τ.

Stock returns R_i and social media opinion scores O_i each amounted to a time series of length equals to 1251 trading days. These series were divided time-wise into $M = 225$ windows $t = 1, 2, \ldots, M$ of width $T = 126$ trading days. A window step length parameter of $\delta T = 5$ trading days defined the displacement of the window, i.e., the number of trading days between two consecutive windows. The choice of window width T and window step δT is arbitrary, and it is a trade-off between having analysis that is either too dynamic or too smooth. The smaller the window width and the larger the window steps, the more dynamic the data are.

To characterize the synchronous time evolution of assets, we used equal time Kendall's rank coefficients between assets i and j, defined as

$$\rho_{i,j}(t) = \sum_{t'<\tau} sgn(V_i(t') - V_i(\tau))sgn(V_j(t') - V_j(\tau)), \tag{7.1}$$

where t' and τ are time indexes within the window t and $V_i \in \{R_i, O_i\}$.

Kendall's rank coefficients takes into account possible nonlinear (monotonic) relationships. It fulfill the condition $-1 \leq \rho_{i,j} \leq 1$ and form the $N \times N$ correlation matrix $C(t)$ that served as the basis for the networks constructed in this work. To construct the asset-based financial and social networks, we defined a distance between a pair of stocks. This distance was associated with the edge connecting the stocks, and it reflected the level at which they were correlated. We used a simple non-linear transformation $d_{i,j}(t) = \sqrt{2(1 - \rho_{i,j}(t))}$ to obtain distances with the property $2 \geq d_{i,j} \geq 0$, forming a $N \times N$ symmetric distance matrix $D(t)$.

7.2.1 Network Filtering: Asset Graphs

We extract the $N(N-1)/2$ distinct distance elements from the upper triangular part of the distance matrix $D(t)$, which were then sorted in an ascending order to form an ordered sequence $d_1(t), d_2(t), \ldots, d_{N(N-1)/2}(t)$. Since we require the graph to be representative of the market, it is natural to build the network by including only the strongest connections. This is a network filtering procedure that has been successfully applied in the construction of *asset graphs* for the analyses of market structure **??**. The number of edges to include is arbitrary, and we included those from the bottom quartile, which represented the 25% shortest edges in the graph (largest correlations), thus giving $E(t) = \{d_1(t), d_2(t), \ldots, d_{\lfloor N/4 \rfloor}(t)\}$. %The presented mechanism for constructing networks defines them uniquely and, consequently, %no additional hypothesis about graph topology is required.

We denoted $E^F(t)$ as the set of edges constructed from the distance matrix derived from stock returns $R(t)$. The financial network considered is $G^F = (V, E^F)$, where V is the vertex set of stocks.

7.2.2 Network Filtering: MST

7.2.3 Network Filtering: PMFG

7.3 Applications

7.3.1 Industry Taxonomy

7.3.2 Market Volatility

The Jaccard Distance, defined as

$$Jaccard(G^F(t'), G^F(t)) = \frac{\|G^F(t') \cap G^F(t)\|}{\|G^F(t') \cup G^F(t)\|}.$$

7.3.3 Portfolio Construction

Bibliography