Financial Risk Management with Machine Learning

THIS IS A TEMPORARY TITLE PAGE
It will be replaced for the final print by a version
provided by the registrar's office.

Thèse n. XXXX 2022
présentée le XXXX
Collège du Management de la Technologie
Chaire Swissquote en finance quantitative
programme doctoral en finance
pour l'obtention du grade de Docteur ès Sciences
par

Marc-Aurèle Antoine DIVERNOIS

Proposition du jury:

Prof Pierre Collin-Dufresne, président du jury
Prof Damir Filipović, directeur de thèse
Prof Michael Rockinger, rapporteur
Prof Simon Scheidegger, rapporteur
Prof Andreas Fuster, rapporteur

Lausanne, EPFL, 2022

Acknowledgements

I wish to express my deepest gratitude to my supervisor, Professor Damir Filipović, for his availability and continuous support during these five years. This thesis greatly benefited from his countless valuable comments and flow of smart ideas. He guided me and I learned a lot from him about doing research. For this, I am truly grateful.
Special thanks go to Professor Michael Rockinger who convinced me to pursue a doctorate degree. He took me under his wing since my Bachelor studies and helped me becoming the researcher I am today. This thesis would not exist without him.
I would like to acknowledge the rest of my thesis committee, Professors Pierre Collin-Dufresne, Andreas Fuster and Simon Scheidegger for their advice and comments.
I am also thankful to Professors Elena Perazzi and Damien Challet. I was their TA for several years and they helped me become a better teacher.
During these years I had the chance to work with extremely nice and intelligent people. I am thankful to my colleagues and friends Antoine Didisheim and Coralie Jaunin for the many chess games and 'bridge tours' around the campus. I am also very grateful to my longstanding friend Christophe George for the proofreading of this thesis.
I would like to thank my family: my parents Jacques and Sonia, my sister Isabelle and her husband Edward, my nephew Dylan and my fiancée Cathy for their unconditional love and for bearing all my endless monologues about my latest models. I dedicate this thesis to them.

Vevey, August 2022 M.-A. D.

Abstract

This thesis consists of three applications of machine learning techniques to risk management.

The first chapter proposes a deep learning approach to estimate physical forward default intensities of companies. Default probabilities are computed using artificial neural networks to estimate the intensities of the inhomogeneous Poisson processes governing default process. The major contribution to previous literature is to allow the estimation of non-linear forward intensities by using neural networks instead of classical maximum likelihood estimation. The model specification allows an easy replication of previous literature using linear assumption and shows the improvement that can be achieved.

The second chapter, titled 'Causal Networks with Neural Networks' is a co-authored work with Damir Filipović (SFI & EPFL), Negar Kiyavash (EPFL) and Jalal Etesami (EPFL). We develop a data-driven framework to identify the interconnections between firms using an information-theoretic measure. This measure generalizes Granger causality and is capable of detecting nonlinear relationships within a network. Moreover, we develop an algorithm using recurrent neural networks and Granger causality to identify the interconnections of high-dimensional nonlinear systems. The outcome of this algorithm is the causal graph encoding the interconnections among the firms. These causal graphs can be used as preliminary feature selection for another predictive model or for systemic risk management. We evaluate the performance of our algorithm using both synthetic linear and nonlinear experiments and apply it to the daily stock returns of US listed firms and infer their interconnections from 1990 to 2020.

The third chapter, titled 'StockTwits Classified Sentiment and Stock Returns' is a co-authored work with Damir Filipović (SFI & EPFL). We classify the sentiment of a large sample of StockTwits messages as bullish, bearish or neutral, and create a stock-aggregate daily sentiment polarity measure. Polarity is positively associated with contemporaneous stock returns. On average, polarity is not able to predict next-day stock returns. But when we focus on specific events, defined as sudden peaks of message volume, polarity has predictive power on abnormal returns. Polarity-sorted portfolios illustrate the economic relevance of our sentiment measure.

Keywords: Risk management, machine learning, neural networks, asset pricing, big data, alternative data.

Résumé

Cette thèse est composée de trois applications de techniques d'apprentissage automatique à la gestion des risques.

Le premier chapitre propose une approche d'apprentissage automatique profond pour estimer les probabilités physiques de défaut des entreprises. Les probabilités de défaut sont calculées en utilisant des réseaux de neurones artificiels pour estimer les intensités des processus de Poisson non-homogènes qui gouvernent les processus stochastiques de défaut. La contribution majeure apportée à la littérature existante est de rendre possible l'estimation non-linéaire des intensités en utilisant les réseaux de neurones artificiels au lieu de la classique estimation du maximum de vraisemblance. Les propriétés du modèle autorisent une réplication aisée de la littérature existante (qui utilise l'hypothèse de linéarité de l'intensité) et montre l'amélioration qui peut être obtenue.

Le deuxième chapitre, intitulé 'Causal Networks wih Neural Networks' est un travail conjoint avec Damir Filipović (SFI & EPFL), Negar Kiyavash (EPFL) et Jalal Etesami (EPFL). Nous développons un modèle axé sur les données et reposant sur une mesure d'information théorique pour identifier les interconnexions entre les entreprises. Cette mesure utilise la causalité de Granger et est capable de détecter des relations non-linéaires à l'intérieur d'un réseau. De plus, nous développons un algorithme qui utilise les réseaux de neurones récurrents ainsi que la causalité de Granger pour identifier les interconnexions dans les systèmes non-linéaires à haute dimension. Le résultat de cet algorithme est le diagramme causal encodant les interconnexions des entreprises. Ces diagrammes causaux peuvent être utilisés comme modèles préliminaires de sélection de variables d'un autre modèle de prédiction ou pour la gestion de risque systémique. Nous évaluons en premier lieu la performance de notre algorithme en utilisant des expériences synthétiques linéaires et non-linéaires puis nous appliquons notre modèle aux rendements journaliers d'actions américaines cotées pour en déduire leurs interconnexions de 1990 à 2020.

Le troisième chapitre, intitulé 'StockTwits Classified Sentiment and Stock Returns' est un travail conjoint avec Damir Filipović (SFI & EPFL). Nous classifions le sentiment d'un large échantillon de messages provenant de StockTwits dans la classe haussière, baissière ou neutre pour créer des séries temporelles de polarité propres à chaque entreprise. La polarité est associée positivement aux rendements d'actions contemporains. En moyenne, la polarité n'est

pas capable de prédire les rendements du jour suivant mais lorsque nous nous focalisons sur des évenements spécifiques, définis par une augmentation soudaine du volume de messages, la polarité a de la puissance prédictive sur les rendements anormaux. Nous illustrons la pertinence économique de notre mesure de sentiment avec des portefeuilles triés par polarité.

Mots-clés: Gestion des risques, apprentissage automatique, réseaux de neurones artificiels, évaluation d'actifs, mégadonnées, données alternatives.

Contents

List of Figures

List of Tables

Introduction

Improved computational power, the rise of Big Data and recent developments in machine learning have created new areas of research in finance. This thesis consists of three machine learning applications to risk management. The first two chapters use neural networks to mitigate default and systemic risk and the last chapter is a financial sentiment analysis using natural language processing.

The first chapter builds on the works from Duffie et al. (2007) and Duan et al. (2012). Both models use a doubly stochastic argument to derive multi-period default probabilities. In particular, they estimate the intensities of two Poisson processes, one governing default and the other governing other exits. Duffie et al. (2007) generate future random values for the covariates using a VAR process while Duan et al. (2012) relax this assumption and use forward intensities. The latter specify the intensities as a linear function of state variables and uses maximum likelihood to estimate the parameters. The first chapter of this thesis extends existing literature by removing the assumption of linear intensities and uses artificial neural networks to estimate the intensities of the Poisson processes. Neural networks are well-suited in this framework because they allow an easy replication of the linear formulation in Duan et al. (2012). Increasing the network's width and depth allows for non-linearities and out-of-sample Lorenz Curves show that the neural network's approach outperforms the linear assumption for every horizon. Finally, I show what the most important predictors of default in the short and longer term are.

Interdepencies in a network are at the heart of systemic risk. The second chapter - connected to the first one as another application of neural networks to risk management - employs Granger causality to identify interconnections among a set of institutions. We build an information measure known as directed information (DI) capable of capturing causal relationships in both linear and non-linear systems. The output of this approach is a directed graph that visualizes the interconnections among a set of time series. Computing DI has high computational and sample complexity which makes it not suitable for inferring the causal structure of large networks. To overcome this problem, we develop a novel approach based on recurrent neural networks that reduces the complexity of evaluating DI in high-dimensional settings. We show that our approach performs well both in linear and non-linear simulated environments, then apply it to infer the causal relationships among US firms from 1990 to 2020.

The last chapter uses natural language processing to assess the predictive power of social media on stock returns. We scrape a large sample of messages from Stocktwits, a microblogging platform similar to Twitter but designed for finance professionals. One of the challenges in this context is to create a classifier that understands the vocabulary of the messages posted by the users. After preprocessing steps, we use TFIDF vectorization to compute the importance of each word in a message. This transforms the messages from a sequence of words to a vector of numbers which is in the same dimension as the vocabulary. Next, we build two adversarial logistic regressions using the TFIDF vectors as features and the user-labels as targets. The first (second) classifier sets bullish (bearish) as positive and non-bullish (non-bearish) as negative class. When the models agree, the classification is trivial and when they disagree, we treat the tweet as neutral. This procedure allows us to create an artificial neutral class that absorbs all the tweets that do not convey financial information. Finally, with daily intervals, we aggregate the predicted sentiments per ticker to compute daily polarity time series. We then use the daily volume of messages on a given firm to identify sudden peaks of activity, indicating a firm event. Computing cumulative average abnormal return and cumulative abnormal polarity in a 41 days window centered at the identified event, we show that abnormal polarities have significant predictive power on the type of event. The performance of sentiment-sorted portfolios illustrates the economic relevance of our sentiment measure.

1 A deep learning approach to estimate forward default intensities

1.1 Introduction

The first default prediction models appeared forty years ago with the first generation model presented by Altman (1968). This work led to the so-called Altman Z-score formula which uses accounting data to compute the default probability of a firm in the next two years. However, when used for financial firms, Altman's Z-score formula needs to be used with care because, as I discuss in this chapter, financial firms have to be treated carefully due to their frequent use of off-balance sheet financing. Twenty years later, a second generation of reduced-form models used econometrical tools such as maximum likelihood, probit, and logit regressions. The major drawback of these models is that they do not provide multi-period forecasts. One innovative recent development is the use of a doubly stochastic Poisson intensity model combined with multiple logistic regressions to account for multi-period default probability estimation. This model is proposed by Duffie et al. (2007) in *Multi-period corporate default prediction with stochastic covariates*. Their main contribution over prior work is to take advantage of the explanatory covariates' dynamics in order to estimate the multiperiod likelihood of default. Their model employs firm-specific and macroeconomic data to create a Markov state vector X_t in order to compute independent firm default intensities $\lambda(t)$ and other types of exit intensities $\phi(t)$. The model proposed in Duffie et al. (2007) is the first one capable of multi-period default probability estimation using time dynamics of covariates X_t. The applications of Duffie et al. (2007) are various. We can find them in credit rating by credit rating agencies, banks who want to calculate the minimal amount of capital to be held and other researches analyzing the link between macroeconomic cycles and firm's default probabilities. Covariates used by Duffie et al. (2007) are firm's trailing one-year stock return, Distance-to-Default, trailing one-year return on S&P500 and three-month Treasury bill rate. Estimating their model on US-listed industrial firms between 1980 and 2004, they find that Distance-to-Default and the current state of the economy have a significant impact on default hazard rates.

The two papers closest to this chapter are from Duffie et al. (2007) and Duan et al. (2012). The first model uses the doubly stochastic argument to derive multi-period default probabilities. To do so, it requires strong assumptions (e.g. vector autoregressive process) regarding the behavior of the time series of covariates to generate future random values for the covariates. If the process is misspecified, biases are introduced both in the forecasted covariates and in the future default probabilities. Five years later, Duan et al. (2012) show that we can relax the VAR assumption with the use of forward intensities. Their paper explains how we can reduce biases by projecting current event realizations on past data. For convenience, the authors specify intensities as a linear function of state variables. I wish to extend the latter by removing the assumption of linear intensities and use an artificial neural network to estimate the intensities of the Poisson processes governing both default and other exits. Machine learning techniques for default probabilities estimation are increasingly drawing attention. Altman et al. (2017) test several machine learning models to predict bankruptcy one year prior to the event. They document a substantial improvement in prediction accuracy using the Z-score as well as six complementary financial variables. However, pure data-driven models often lack economic relevance and this is where the forward intensity model can contribute to the literature. The forward intensity model is able to provide multi-period predictions while being supported by an economic and econometric background.

In Duffie et al. (2007), one of the main assumptions is that the covariates governing both default and other exits intensities follow a high-dimensional vector autoregressive (VAR) process. Using this type of process forces the model to greatly reduce either the number of firms in the sample or the number of state variables explaining firm attributes; if one does not restrict the number of firms or variables in the estimation, the dimension of the model will simply be too high and it will considerably increase computational time. A major step forward made in Duan et al. (2012) is to get rid of the VAR process in order to reduce computational time by using a new reduced-form approach based on a forward intensity model. These forward intensities produce a term structure of bankruptcy probabilities without using any sort of high-dimensional process. Using this method allows the model to incorporate a lot more state variables or individual firms in the sample. Moreover, Duan et al. (2012) state that their model may also improve robustness to misspecification because in a VAR model, estimation of future values are highly sensitive to any bias. On the other side, the forward intensity model approach is a projection of past observations on current realizations, which does not involve random estimation of future values.

Regarding covariates used, both Duan et al. (2012) and Duffie et al. (2007) estimate their own Distance-to-Default (hereafter DtD). An important aspect to highlight is that Distances-to-Default specified in Duffie et al. (2007) differ from those estimated in Duan et al. (2012) since the former estimate DtD using the variance restriction method (see (Duan and Wang, 2012)) while the latter use the transformed-data maximum likelihood estimation method to account for financial firms. The variance restriction method is a popular way to implement the Merton (1974) model but fails at estimating properly the default point for financial firms. Following the KMV assumption (see Crosbie and Bohn (2003)), the default point in this method

is specified as short-term debt plus one half of long-term debt and does not take into account other liabilities. However, it is well-known that financial firms such as banks specify a high portion of their debt as other liabilities. Hence, to include financial firms in the sample, the default point has to be adjusted to the sum of short-term debt, one half of long-term debt and a fraction δ of other liabilities. Duan et al. (2012) use a maximum likelihood estimation in order to compute the unknown fraction δ. The Appendix provides additional methodological information on the DtD estimation using the variance restriction method and the maximum likelihood estimation.

To summarize, the forward intensity model requires an assumption to link covariates to intensities. Duan et al. (2012) use a linear assumption and a maximum likelihood to estimate those parameters. This chapter contributes to previous literature by using neural networks to relax the linear assumption of forward intensities. Neural networks allow the estimation of highly non-linear functions without specifying the form of the relationships. The remainder of this chapter is structured as follows. Section 1.2 sets up the reduced-form model of default, develops the likelihood function used as loss function later on and describes the neural network approach. Section 1.3 discusses summary statistics of the dataset used. Section 1.4 presents the results. Section 1.5 concludes. Appendix A.1 at the end of the thesis contains several proofs and more details on the Distance-to-Default estimation.

1.2 Methodology

1.2.1 Default model

The model adds to the literature on reduced-form models of default for multiperiod corporate prediction using the doubly stochastic formulation as in Duffie et al. (2007) and Duan et al. (2012) (Duan henceforth). The default's time is modeled as the stopping time

$$\tau_D = \inf\{t : N_t > 0, M_t = 0\}, \tag{1.1}$$

where N_t and M_t are the counting processes governing default and other exits respectively. Similarly, the stopping time for combined exits is denoted by

$$\tau_C = \inf\{t : N_t > 0 \wedge M_t > 0\}. \tag{1.2}$$

We have the following :

$$\left.\begin{array}{r} \text{if the firm exits due to default, } \tau_{Ci} = \tau_{Di}, \\ \text{if the firm does not exits due to default, } \tau_{Ci} < \tau_{Di}. \end{array}\right\} \quad \Rightarrow \tau_{Ci} \leq \tau_{Di}$$

Let us denote by Z_{it} the set of firm-specific variables at time t for the firm i and Y_t the set of macroeconomic variables at time t. Let t_i^0 be the first time of entry of firm i in the dataset.

The econometrician's information set $\mathscr{F}_t$ at time t is thus

$$\mathscr{F}_t = \{Y_s : s \leq t\} \cup \mathscr{G}_{1t} \cup \mathscr{G}_{2t} ... \cup \mathscr{G}_{Nt}, \tag{1.3}$$

where

$$\mathscr{G}_{it} = \{(1_{\tau_{C_i} < u}, 1_{\tau_{D_i} < u}, Z_{iu}) : t_i^0 \leq u \leq \min(\tau_D, \tau_C, t)\}. \tag{1.4}$$

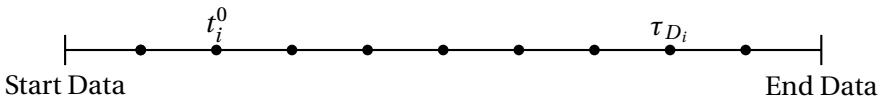


Figure 1.1: Example of the lifespan of a firm
Each dot corresponds to a time period where the econometrician gathers firm-specific (if available) and macroeconomic variables. t_i^0 is the entry time of the firm i and τ_{D_i} denotes the default time of firm i.

In Figure 1.1, each dot corresponds to a period. At each period, the econometrician gathers firm-specific variables (DtD, Cash/Total Assets, Net Income/Total Assets, ...) and macroeconomic variables (S&P500 return, treasury rate). For a particular point in time t, the econometrician knows the time series of macroeconomic variables until t irrespective of t_i^0, and the time series of firm-specific variables from t_i^0 to τ_{D_i} if $t_i^0 \leq t$ and $t \leq \tau_{D_i}$. Following Duffie et al. (2007), the conditional probability of default within s years can be computed as

$$\mathbb{P}[\tau_D < t + s | \mathscr{F}_t] = E_t \left[\int_t^{t+s} e^{-\int_t^z (\lambda(u) + \phi(u)) du} \cdot \lambda(z) dz \right]. \tag{1.5}$$

The probability of default is a function of intensities λ (default) and ϕ (other exits). However, these intensities are unknown and unobservable. In Duffie et al. (2007), the state variables governing Poisson intensities are assumed to follow a specific vector autoregressive (VAR) process. This assumption is relaxed in Duan's paper by using forward intensity rates. Instead of modeling λ_{it} and ϕ_{it} as some functions of state variables available at time t, Duan et al. (2012) propose to deal with $f_{it}(\tau)$ and $g_{it}(\tau)$ directly as functions of state variables available at time t and the forward starting time of interest τ. The analogy to interest rates would be that λ_t is the short rate and $f_t(u)$ is the forward rate for horizon u. Duan et al. (2012) propose a model to predict corporate defaults at multiple horizons by estimating these forward intensities via maximum likelihood. To do so, they use a linear assumption in the relationship between the variables and the forward intensities (i.e. $f_{it}(\tau) = \exp(\alpha_0(\tau) + \alpha_1(\tau) x_{it,1} + \alpha_2(\tau) x_{it,2} + ... + \alpha_k(\tau) x_{it,k})$. However, it is highly likely that default intensities depend on those covariates in a non-linear way. I propose to use an artificial neural network (ANN) to find the set of weights governing the process f_{it} and g_{it}. I show that I am able to capture potential non-linear relationships between the state variables and the forward intensities, which significantly improve forecasts.

Forward intensities

Since we do not have the exact knowledge of λ and ϕ, Duan et al. (2012) propose to use forward intensity rates. However, the default probability given in equation 1.5 (which depends on spot intensities) needs to be translated into a formula that depends on forward intensities. To do so, we first compute the survival probability as a function of forward combined intensity, which will later be used to compute the probability of default as a function of both combined and default forward intensities. Let us denote $F_{it}(\tau)$ the conditional distribution function of the combined (default and other exits) exit time evaluated at $t+\tau$. Hence, 1-$F_{it}(\tau)$ is the probability of surviving in the interval $[t, t+\tau]$. Therefore, we have :

$$1 - F_{it}(\tau) = \mathbb{E}[e^{-\int_t^{t+\tau}(\lambda(s)+\phi(s))ds}]. \tag{1.6}$$

Next, let us introduce the quantity $\psi_{it}(\tau)$ to be :

$$\psi_{it}(\tau) \equiv -\frac{\ln(1-F_{it}(\tau))}{\tau} \equiv -\frac{\ln(\mathbb{E}[e^{-\int_t^{t+\tau}(\lambda(s)+\phi(s))ds}])}{\tau}. \tag{1.7}$$

Reverting equation 1.7 gives :

$$e^{-\psi_{it}(\tau)\cdot\tau} = 1 - F_{it}(\tau). \tag{1.8}$$

Where $e^{-\psi_{it}(\tau)\cdot\tau}$ is again the survival probability. We now need to compute $\psi_{it}(\tau)\cdot\tau$. At this point, Duan et al. (2012) make the assumption that ψ_{it} is differentiable and define the forward combined exit intensity as

$$g_{it}(\tau) \equiv \frac{F'_{it}}{1-F_{it}}. \tag{1.9}$$

Equation 1.9 comes from the definition of a hazard rate function. Referring Schönbucher (2003), the definition of a hazard rate function is the following:

Definition 1.2.1 (Hazard rate)**.** Let τ be a stopping time and $F(T) \equiv \mathbb{P}[\tau \leq T]$ its distribution function. Assume that $F(T) < 1 \forall T$, and that F(T) has a density f(T). The hazard rate function h of τ is :

$$h(T) \equiv \frac{f(T)}{1-F(T)}.$$

A hazard rate is the local arrival probability of a stopping time per time interval. Under suitable regularity conditions, intensities and hazard rates are closely similar. In particular, in our doubly-stochastic framework, hazard rates and intensities are equivalent. Hence, in Duan et al. (2012) and this chapter, $\lambda(t) = h(t)$ and thus the distinction between hazard rates and intensity is not made.

The relation between $g_{it}(\tau)$ and $\psi_{it}(\tau)$ is given by :

$$\begin{aligned} g_{it}(\tau) &= \frac{F'_{it}(\tau)}{1-F_{it}(\tau)} \\ &= \psi_{it}(\tau) + \psi'_{it}(\tau)\tau. \end{aligned} \tag{1.10}$$

Next, we can compute[1] the quantity $\psi_{it}(\tau)\tau$ that we were looking for as :

$$\psi_{it}(\tau)\tau = \int_0^\tau g_{it}(s)ds. \tag{1.11}$$

Hence, the probability of surviving over [t,t+τ] is given by :

$$\mathbb{P}[\tau_c > t+\tau | \mathbb{F}_t] = \exp(-\int_0^\tau g_{it}(s)ds). \tag{1.12}$$

The forward default intensity for horizon τ is defined as the limit for a small time step of the probability of defaulting in this small time step given that the firm survives until the considered horizon. The probability is Bayesian and the forward default intensity denoted $f_{it}(\tau)$ is the following :

$$f_{it}(\tau) \equiv \frac{\lim_{\Delta t \to 0} \frac{\mathbb{P}[t+\tau < \tau_{Di} = \tau_{Ci} \leq t+\tau+\Delta t]}{\Delta t}}{e^{-\psi_{it}(\tau)\cdot\tau}}. \tag{1.13}$$

Hence, the probability of defaulting between t and $t+\tau$ is given by :

$$\int_0^\tau e^{-\psi_{it}(s)s} f_{it}(s)ds = \int_0^\tau e^{-\int_0^s g_{it}(u)du} f_{it}(s)ds. \tag{1.14}$$

Likelihood function

In this setup, the likelihood function depends on three types of probabilities (default, other exit and surviving) which themselves depend on two types of intensities (default and other exits). The negative log-likelihood function has to be adjusted to the neural network framework and can be used as an objective function to be minimized by feeding batches of data points. Batch feeding is a common practice in machine learning and consists in splitting the available data in subsets of fixed size. Next, each backward pass takes one batch to perform a gradient descent to update the parameters of the model.

[1] Proofs of the following formulations can be found in Appendix A.1.

To allow further comparison with Duan et al. (2012), I employ the same discretization of time: $t = 0, 1, 2, ...$ and $\tau = 0, 1, 2, ...$ are time sequences of one month increments. Similarly, $f_{it}(\tau)$ and $g_{it}(\tau)$ are forward intensities computed at time t for the period [t+τ, t+τ+1]. The use of the τ index is to account for multiperiod prediction. When $\tau = 0$, the forward intensity model computes spot intensities. When we set $\tau = 1$, the forward intensity model produces estimates one step ahead, and so forth. I denote $X_{it} = (x_{it,1}, x_{it,2}, ...)$ the set of firm-specific and macroeconomic variables explaining both default and combined exit intensities. As specified in Duan et al. (2012), $f_{it}(\tau)$ and $g_{it}(\tau)$ are functions of X_{it} and can be specified as any form of function as long as they satisfy the constraints that follow. Since combined exit intensity has to be greater than or equal to default intensity, we need to make sure that the forms specified for $f_{it}(\tau)$ and $g_{it}(\tau)$ satisfy the following conditions : $f_{it}(\tau) \leq g_{it}(\tau)$, $f_{it}(\tau) > 0$, $g_{it}(\tau) > 0$.

I design two neural networks. One is trained to compute f_{it} and the other is trained to output h_{it} where $g_{it} = f_{it} + h_{it}$. I impose non-negativity on outputs of both models such that the combined exit intensity will never be smaller than the default intensity for all horizons. Let us denote λ and μ the set of parameters (weights) tuned in the neural network for f_{it} and h_{it} respectively. $N^{(\lambda)}$ and $N^{(\mu)}$ represent the output of the neural network for f_{it} and h_{it} respectively. The log-likelihood for prediction horizon τ is expressed[2] as

$$\mathscr{L}(\lambda(s)) = \sum_{i=1}^{N} \sum_{t=0}^{T-s-1} \mathscr{L}_{i,t}(\lambda(s)), \qquad s = 0, 1, ..., \tau - 1 \tag{1.15}$$

$$\mathscr{L}(\mu(s)) = \sum_{i=1}^{N} \sum_{t=0}^{T-s-1} \mathscr{L}_{i,t}(\mu(s)), \qquad s = 0, 1, ..., \tau - 1 \tag{1.16}$$

where

$$\begin{aligned} \mathscr{L}_{i,t}(\lambda(s)) = & \underbrace{\mathbf{1}_{t_{0i} \leq t, \tau_{Ci} > t+s+1}}_{(1)} \cdot (-N_{it}^{(\lambda)}(s)) \Delta t) \\ & + \underbrace{\mathbf{1}_{t_{0i} \leq t, \tau_{Di} = \tau_{Ci} \leq t+s+1}}_{(2)} \cdot \ln(1 - \exp[-N_{it}^{(\lambda)}(s) \Delta t]) \\ & + \underbrace{\mathbf{1}_{t_{0i} \leq t, \tau_{Di} \neq \tau_{Ci}, \tau_{Ci} \leq t+s+1}}_{(3)} \cdot (-N_{it}^{(\lambda)}(s) \Delta t), \end{aligned} \tag{1.17}$$

$$\begin{aligned} \mathscr{L}_{i,t}(\mu(s)) = & \underbrace{\mathbf{1}_{t_{0i} \leq t, \tau_{Ci} > t+s+1}}_{(1)} \cdot (-N_{it}^{(\mu)}(s) \Delta t) \\ & + \underbrace{\mathbf{1}_{t_{0i} \leq t, \tau_{Di} \neq \tau_{Ci}, \tau_{Ci} \leq t+s+1}}_{(3)} \cdot \ln(1 - \exp(-N_{it}^{(\mu)}(s)) \Delta t)). \end{aligned} \tag{1.18}$$

[2] Proof of the above formulation can be found in Appendix A.1.

The likelihoods (1.17) and (1.18) are the sum of the products between event indicator functions and their respective occurence probability. I define the indicator function $\mathbf{1}_{A<B}$ to be one if $A < B$ or zero otherwise. These likelihoods specify three mutually exclusive indicator functions, defining three cases over the time interval $[t, t+\tau+1]$:

1. The firm does not exit the sample between t and $t+\tau$ and is classified as surviving. This case is specified as (1) in the likelihood because the combined exit time τ_{Ci} is not in the interval $[t, t+\tau+1]$.

2. The firm defaults and exits the sample during the interval. This case is specified as (2) because $\tau_{Ci} = \tau_{Di}$ when the firm exits due to default jointly with τ_{Di} being in the interval $[t, t+\tau+1]$.

3. The firm exits the sample for other reasons. This case is specified in (3) since the stopping time $\tau_{Di} \neq \tau_{Ci}$ jointly with $\tau_{Ci} \leq t+\tau+1$.

As in previous studies, the likelihood functions still exhibit the decomposable property which allows to estimate the model for each horizon of prediction independently.

Since the intensities are directly driven by the covariates, Duan et al. (2012) require an assumption on the mapping from the covariates to these intensities. In Duan et al. (2012) the mapping is made with a linear assumption, whereas in the framework of this chapter, the mapping depends on the whole architecture of the neural network. When the neural network has only one hidden layer of one neuron coupled with an exponential activation function, the model boils down to Duan et al. (2012) since the intensities would be a linear combination of the covariates. As the width and depth of the network increases, we depart more and more from the linear assumption and we allow more non-linearities to be incorporated in these intensities.

1.2.2 Neural Networks

Neural networks can be seen as a very general function to map a given input (in this case firm-specific and macroeconomic variables) into a desired output (forward intensities). They learn how to compute the output by tuning weights in order to minimize a given loss function. A neural network is constructed by juxtaposing several hidden layers of neurons. The input of each layer is a data transformation of the output of the previous layer. Initially, the weights of the network are assigned random values. Then, the training process starts and consists of many iterations of a forward pass and a backward pass. The forward pass takes as input a batch of data and computes the loss value; the backward pass then computes the gradient and adjusts the weights of the network based on a learning rate hyperparameter. As an illustration, Figure 1.2 shows a neural network with 2 hidden layers : 5 neurons in the first layer and 3 neurons in the second layer. In this example, 3 features (inputs) are fed to the network.

Each feature is connected to the first hidden layer by a set of weights. The outputs of the first layer are also weighted to produce the inputs of the second hidden layer. Non-linearity is introduced in each node by a non-linear activation function (e.g. sigmoid). Finally, the output of the second hidden layer is aggregated to produce the final output of the model. The neural networks in this chapter are implemented in Python using the library TensorFlow. Approximately one hour of computing time is needed to fit all networks on a 32GB RAM quad core 2.7 GHz computer.[3]

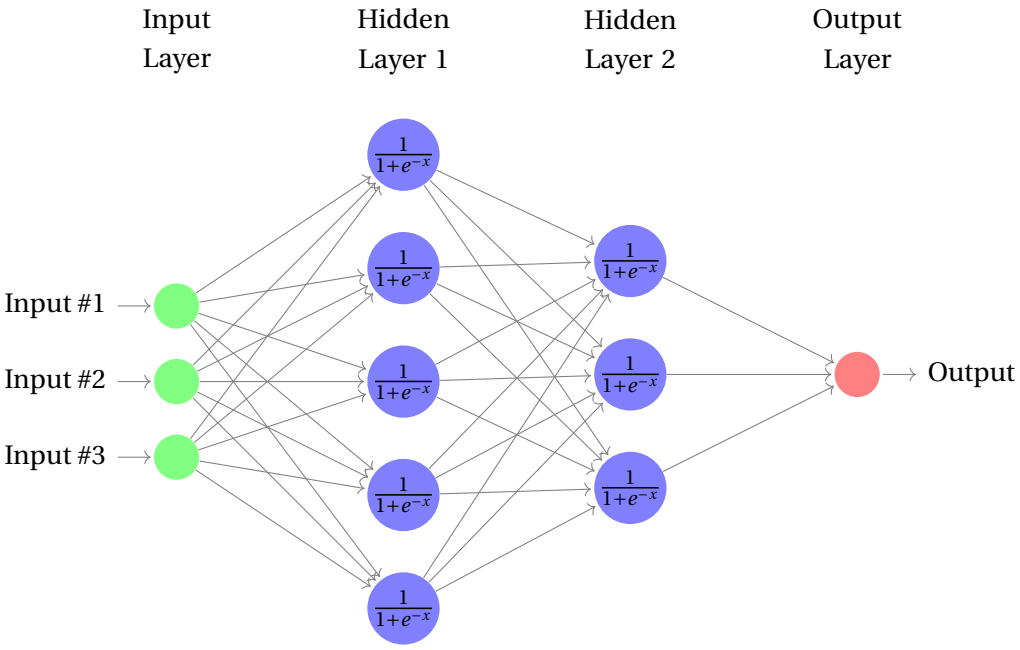


Figure 1.2: Illustration of a neural network [5, 3]
In this network, there are three input features (green circles), five neurons in the first hidden layer and 3 neurons in the second hidden layer. Each neuron is activated with a sigmoid function.

The use of neural networks can be motivated twofold. First, neural networks are well suited to approximating a function (in our case forward intensities) with the advantage of having different degrees of modularity. By definition, the architecture of the network generates the form of the function approximated. A deeper network allows for more non-linearities in the approximation of the function, at the expense of having more parameters to estimate. For instance, suppose that every observation comes with 12 input features (i.e. x is a vector of shape 1x12), a neural network [5, 3] (i.e. 2 layers neural network with 5 neurons in the first hidden layer and 3 neurons in the second hidden layer) can be viewed as a function computing the output $N_{it}^{(\lambda)}$ in the following way :

$$N_{it}^{(\lambda)} = \Big[\phi_1(\Big[\phi_2(\Big[x\Big]_{1\times 12}\Big[w_1\Big]_{12\times 5} + \Big[b_1\Big]_{1\times 5}\Big)\Big]_{1\times 5}\Big[w_2\Big]_{5\times 3} + \Big[b_2\Big]_{1\times 3}\Big)\Big]_{1\times 3}\Big[w\Big]_{3\times 1},$$

[3] GPU computing is not necessary in this context as the networks are small.

with the activation functions being for instance the sigmoid function $\phi_1(x) = \phi_2(x) = \frac{1}{1+e^{-x}}$, x being the data input, w_1, w_2 and w weight matrices and b_1 and b_2 biases matrices. In this architecture, the number of parameters to estimate is equal to 12x5 + 1x5 + 5x3 + 1x3 + 3x1 = 86.
Second, neural networks are well-suited for this chapter because they allow an easy replication of the benchmark model Duan et al. (2012). More specifically, if the activation function $\phi(x)$ is chosen as being an exponential $\exp(x)$, and the network architecture is [1] (i.e. a single hidden layer with a single neuron), the output $N_{it}^{(\lambda)}$ becomes the linear assumption

$$\begin{aligned} N_{it}^{(\lambda)} &= \phi(\left[x\right]_{1\times 12} \cdot \left[w\right]_{12\times 1} + \left[b1\right]_{1\times 1}) \\ &= \exp(b1 + w_1 \cdot x_1 + w_2 \cdot x_2 + ... + w_{12} \cdot x_{12}), \end{aligned}$$

The results from this architecture are described in Section 4.

1.3 Empirical section

1.3.1 Data

The accounting data is taken from the Wharton Research Data Services (WRDS) using the CRSP/Compustat merged database. The macroeconomic data is taken from CRSP, the Federal Reserve Bank Reports and Datastream. The bankrupcty data is taken from the Compustat database, using the DLRSN item for the reason of deletion and the DLDTE item for the date of deletion. DLRSN contains the code that indicates the reason why a company becomes inactive on the database. I consider firms with a DLRSN code 2 (bankruptcy) or 3 (liquidation) to be defaulted, any other DLRSN code as other exits and no DLRSN code as surviving. For additional information on DLRSN and DLDTE codes, I refer to the Wharton WRDS documentation. I focus on the period from 1991 to 2018 to match the accounting data with the bankruptcy data. Using the WRDS database, I download accounting information for every company that has been listed someday on either NYSE, AMEX or NASDAQ between 1991 and 2018. The dataset spans 27 years of data where firms entered and/or exited anywhere in this sample. Using this kind of sample brings a problem of cylindric data : firm's entering/exiting time are not the same for each company. The dataset is represented as a three dimensional matrix with the x-axis being features (i.e. variables), the y-axis being time, and the z-axis being firms. I fill the matrix with missing values for elements where firm i does not exist or already left at time t. Since neural networks need a lot of data points to be well trained, I choose not to remove firms even if they have a short lifespan. When a firm has a variable completely missing, I drop the whole firm because the likelihood is not specified if a variable is fully missing. However, when the variable is not fully missing but only some data points are not available, I use the last available information before the missing entry. I winsorize all variables at the 2.5 and 97.5

Horizon	Survivals	Defaults	Other exits
0	2'025'094	514	10'746
3	1'972'063	499	10'713
6	1'918'061	499	10'392
12	1'811'323	454	10'393
24	1'610'242	382	9'193
35	1'457'240	343	8'295

Table 1.1: Number of observations in each category for each horizon of prediction τ
These correspond to the firm-month observations used in the likelihood for horizon τ.

percentile. Finally, I standardize all variables by substracting the mean of the variable and dividing the result by its standard deviation. The variables in the test set are also standardized using their respective mean and standard deviation from the training set. Table 1.1 shows the number of firms in the three categories for each horizon of prediction τ, which corresponds to the firm-month observations used in the likelihood for horizon τ (see equations 1.17 and 1.18). Figure 1.3 shows the total number of firms that defaulted, survived or exited for other reasons plotted on a year on year basis.

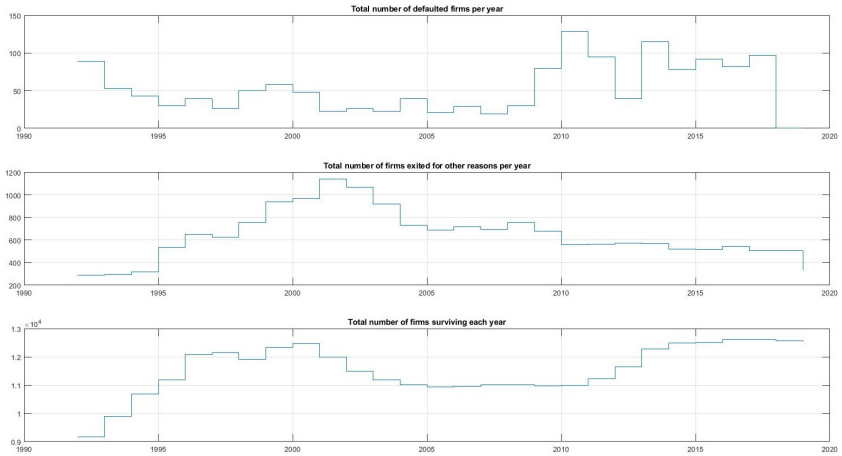


Figure 1.3: Defaults, exits for other reasons and survivals each year
The top plot shows the number of defaulted firms, the middle plot shows the number of exits for other reasons and the bottom plot shows the number of firms surviving each year.

Leippold et al. (2012) use a theoretical model to show that the most powerful default predictor must incorporate both macroeconomic and accounting data. For more transparency and to allow better comparison with previous literature, I choose to work with a similar set of features than the benchmark model Duan et al. (2012). Firm-specific values are common to each firm, macroeconomics variables are function of market data. The exhaustive list of variables is the following :

1. SP500 : trailing 1-year return on the S&500 index.
2. Treasury : 3-month annualized US Treasury bill rate. I use this variable as the risk-free rate r in the model.
3. CASH/TA : cash and short-term investment divided by total assets. Both quantities are taken from the balance sheet of the company.
4. NI/TA : net income divided by total assets.
5. SIZE : log of the market equity value divided by the cross-sectional average market equity value. The number of shares outsanding times the stock price gives the market equity value.
6. DtD : the Distance-to-Default is a measure introduced in Merton (1974) to gauge how close a firm is from default. In this chapter, the DtD is estimated using a maximum likelihood taking into account other liabilities of each firm to handle financial firm's bias. It is well-known in the literature that DtD is a significant measure to estimate default probabilities but has to be used jointly with other variables. See the Appendix A.1 for additional information regarding the estimation procedure of DtD.
7. M/B : asset market value divded by the total book asset value.

To capture momentum of variables, I also compute one step rolling window differences for each firm-specific variable. These variables are called "Δ" followed by the name of the variable. They represent whether the firm has been improving or deteriorating with respect to this particular variable comparing to the last period performance. Given this model specification, the Δ is particularly interesting because if a firm shows many consecutive negative delta values, it means the company is in danger. However, it is also important to look at the level value to compare a defaulted firm with a non-defaulted firm. The intuition tells us that prior default time, a defaulted company should have shown lower level values (for instance, DtD) than a non-defaulted firm.

1.3.2 Summary statistics

This section depicts the summary statistics of the dataset. Table 1.2 shows a summary of the mean of each variable for the three categories of firms. Please note that we must be very careful when comparing two means in this table. Comparing means needs to be done with confidence intervals and significance tests, which involves standard deviations. This table is presented to give a rough idea without performing any statistical inference. The table shows that the average defaulted firm is smaller, has a lower DtD, a smaller Market-to-book ratio, loses more money and has less cash than the average surviving firm. The prefixes Δ in front of the variables stand for the one-lag differences. Finally, Figure 1.4 shows the correlation matrix for the twelve covariates.

	Surviving	Default	Other exits
Cash/TA	0.1952	0.1896	0.1851
NI/TA	-0.0756	-0.2057	-0.129
Size	-2.7845	-4.6743	-3.621
DtD	10.641	6.441	8.401
MBratio	2.3774	1.6995	2.202
Δ CASH/TA	-0.0026	0.0023	-0.0074
Δ NI/TA	-0.0019	-0.0433	-0.0148
Δ Size	-0.0305	-0.5095	-0.0939
Δ DtD	-0.1366	-0.7346	0.0295
Δ MBratio	-0.0285	0.0520	0.0801

Table 1.2: Mean of variables for surviving firms, defaulted firms and other exits
The prefix Δ stands for a one period lagged difference.

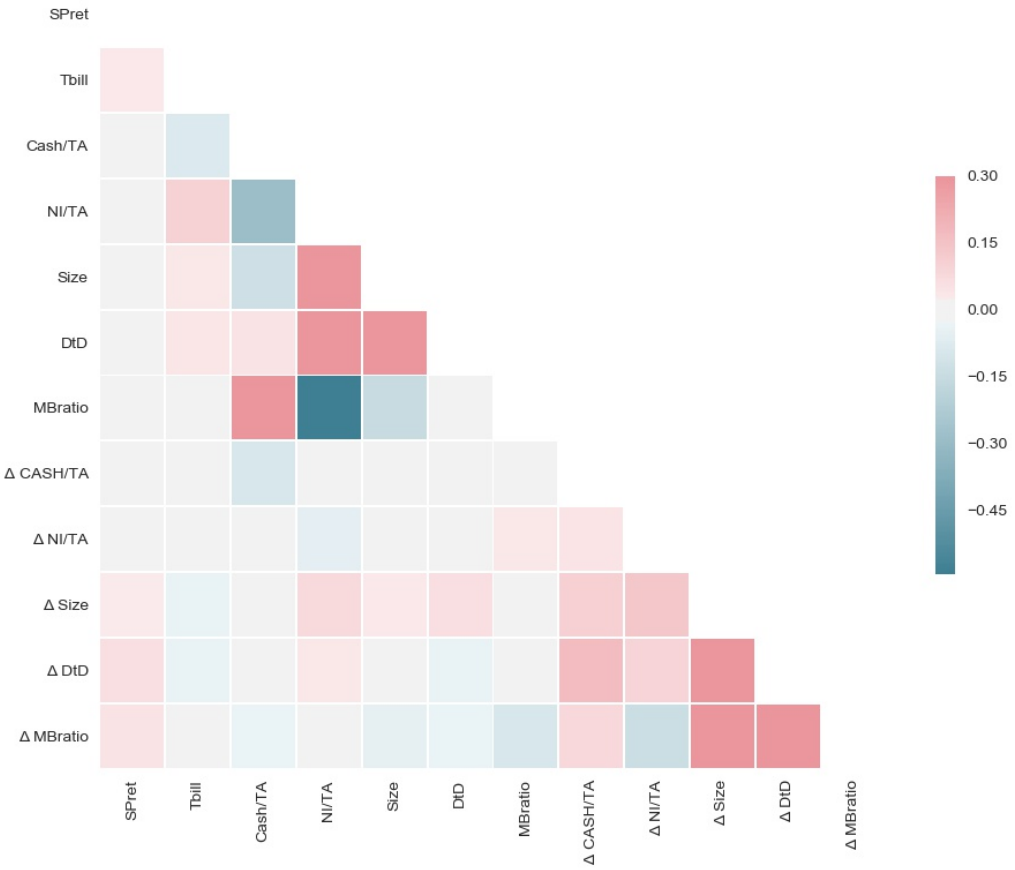


Figure 1.4: Correlation matrix for firm-specific and macroeconomic covariates
Red values represent positive correlation coefficients, blue values shows negative correlation coefficients.

1.4 Results

As for every neural network, we need to chose the optimal hyperparameters to achieve the highest performance. In my setup, this consists mainly of choosing the architecture of the model (i.e. the number of neurons and layers). To do so, there is currently no other better method than trial and error. However, hyperparameters need to be carefully chosen to not overfit the test set. To avoid any overfitting, I perform a 5-fold cross validation for each horizon of prediction. I cut 15% of the dataset as test set, all results that I will talk about in this chapter are out-of-sample and performed on the observations from the test set that the model has never seen before. The remaining set of observations is partitioned into smaller subsets so that in every fold of the validation a different subset is used as validation set and the rest is used as training set. Finally, to measure the discriminatory power of the models, I use the Lorenz curve (Lorenz (1905)) and I use the Gini coefficient as a scalar performance measure to aggregate across folds to get the measure that I use to discriminate the models (see Definition 1.4.1 (Leippold et al. (2012))).

Definition 1.4.1 (Lorenz curve)**.** The Lorenz curve of a predictor P is the two-dimensional graph

$$(\mathbb{P}\{P \leq p\}, \mathbb{P}\{P \leq p | Y = 1\}),$$

$\forall\, p \in (-\infty, +\infty)$.

The Lorenz curve plots on the x-axis the cumulative percentage of observations against the fraction of defaults on the y-axis. These curves[4] are often used in the literature for default prediction (for instance Leippold et al. (2012), Duan et al. (2012)). They are different from ROC curves and precision-recall curves because they do not rely on thresholds to discriminate between true positives and false positives. Hence, they are particularly well-suited as performance measure in this model because of the multiperiod framework involved. The idea is that if the model is outputting a false positive for an horizon τ but the true positive is horizon $\tau + 1$, the model should not be too hardly penalized. The ROC and precision-recall curves would treat this as a false positive even though the prediction was not that far from the target. The idea behind the Lorenz curve is to order default probabilities and look how they are distributed across defaulted and non-defaulted firms. We can easily see whether the model is outputting high probabilities for defaulted firms and small probabilities for surviving firms.

Finally, the Gini coefficient is used as the scalar summary statistic to compare the models. It measures the degree of inequality of the Lorenz curve. A perfect model has a Gini coefficient close to 1 and a poor model has a Gini coefficient of 0 (perfect equality).

[4] Similar plots exist also under different names (e.g. power curves, cumulative accuracy profiles).

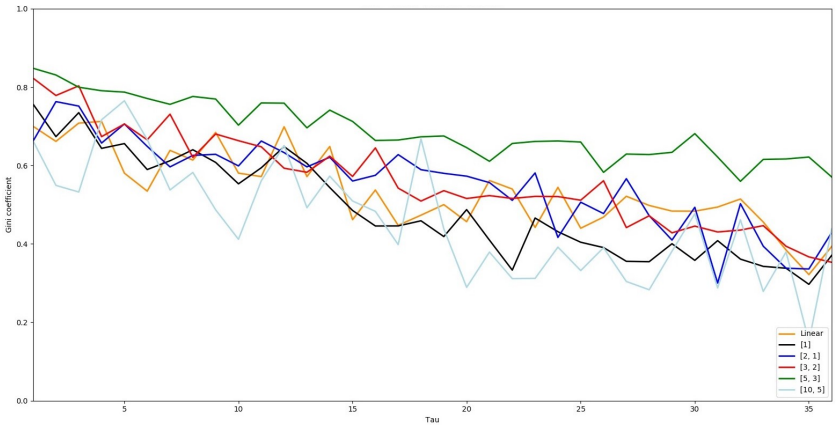


Figure 1.5: Gini coefficient : Linear vs Neural Networks
The yellow line shows the Gini coefficient of the linear model for every horizon. The remaining lines show the Gini coefficient for different neural network architectures. The green line is the best performing model as it exhibits the highest Gini for every horizon.

1.4.1 Model choice

The choice of the optimal architecture of the neural network is made using k-fold cross-validation. The average Gini coefficient across all folds for every horizon is used as a comparison tool to determine the best model architecture. If the architecture is too deep, the implicit function computed in the network to output forward intensities incorporates too many parameters and the risk of overfitting is larger, resulting in a lower accuracy measure. Similarly, if the network is not deep enough, the forward intensities are computed using a too simplistic representation and result in a low accuracy measure as well. This is usually known in the machine learning literature as the bias-variance tradeoff. Figure 1.5 shows the average Gini coefficient computed on the validation test across all folds of the cross validation for each horizon. I compare the scores obtained with different network architectures with those using the linear assumption. In this setup, one can interpret the architecture model as the non-linearity degree because a higher architecture involves more weights in the output function to be estimated. By disantangling the spaghettis, we can clearly see the bias-variance tradeoff. Increasing the depth of the networks from the simplest neural network [1] to a two layers network [2, 1] increases the Gini coefficients, in particular at mid horizons. This is probably due to the non-linearities introduced via the second layer. Next, it looks like the [3, 2] performs similarly as the [2, 1]; but we clearly see that [5, 3] dominates any other previous models. Finally, increasing the depth again to [10, 5] decreases Gini coefficient at all horizons, suggesting a severe overfitting of the forward intensities function. For the following sections, I will now only show out-of-sample results of the [5, 3] architecture using the test set.

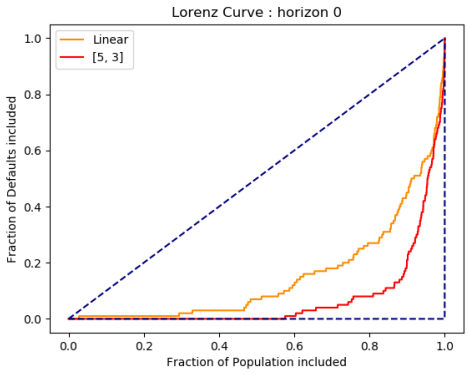
Horizon	[5, 3]	Linear
0	0.85	0.70
3	0.79	0.71
6	0.76	0.64
12	0.70	0.57
24	0.66	0.44
35	0.57	0.39

Table 1.3: Gini coefficients
Comparison of the out-of-sample Gini coefficients associated to the [5, 3] neural network and the linear assumption.

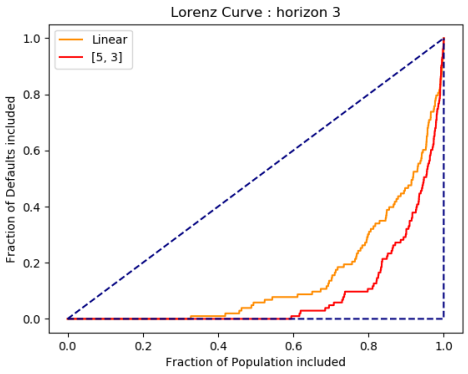
1.4.2 Performance

Figure 1.6 shows Lorenz curves for horizons 0, 3, 6, 12, 24 and 35 but all results generalize well to all other horizons. The curves are completely out-of-sample since they are computed on the test set that the model has never seen before. Table 1.3 shows Gini coefficients for both the linear assumption Duan et al. (2012) and for the [5, 3] for the same horizons. Overall, as expected the neural network clearly outperforms the linear assumption, suggesting that the linear assumption from Duan et al. (2012) can be greatly improved by adding non-linearities in the specification of the intensities. Unfortunately, it is difficult to tell which kinds of non-linearities should be taken into account since neural networks are often seen as a black box. However, I will still try to answer this question by looking how the average intensity outputted by the model changes when we change a feature ceteris paribus (see Section 1.4.4 dedicated to sensitivities). Another attempt at answering this question is described in Section 1.4.3, where I dive into the weights of the network to understand how the output is computed.

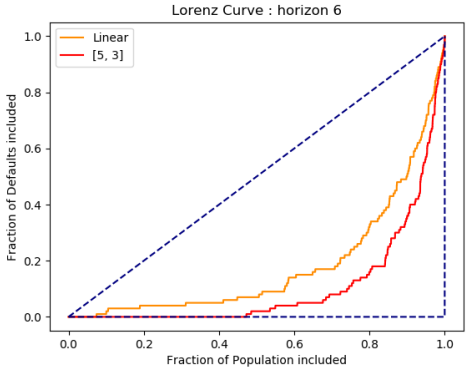
For comparison purposes, Figure 1.7 exhibits the Lorenz curves of the linear assumption, the replication of the linear assumption in the neural network framework, and the [5, 3] model. The linear assumption and its replication have a similar performance, showing that the neural network “NN+exp+[1]" is indeed able to replicate the linear framework depicted in Duan et al. (2012).



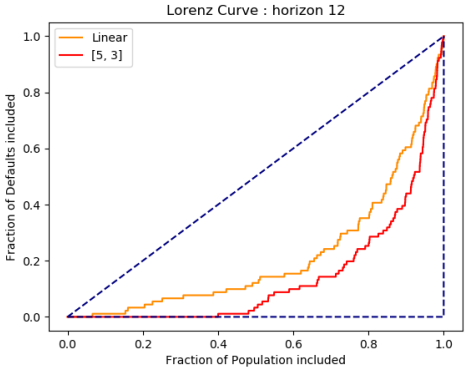
Horizon 0



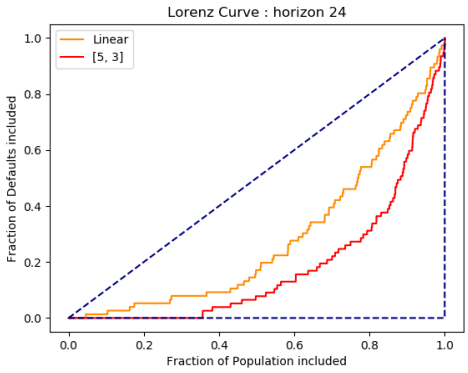
Horizon 3



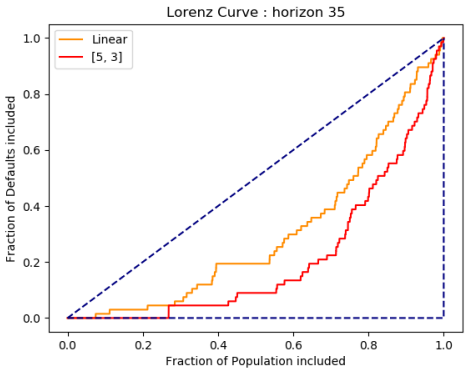
Horizon 6



Horizon 12



Horizon 24



Horizon 35

Figure 1.6: Out-of-sample Lorenz Curves for each horizon
The orange line shows the performance of Duan's model and the red line shows the performance of the [5, 3] neural network.

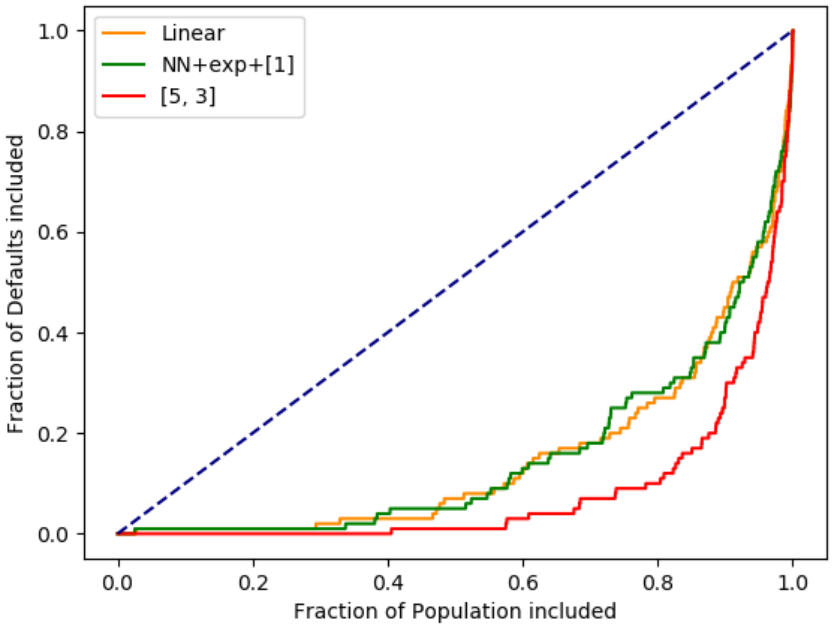


Figure 1.7: Comparison with the benchmark model Duan et al. (2012)
The green line shows the performance of the [1] neural network (i.e. a single hidden layer with a single neuron) with an exponential activation function. The orange line shows the performance of Duan's model and the red line shows the performance of the [5, 3] neural network.

1.4.3 Computational graph

Figure 1.8 is a representation of a fully trained neural network for the forward default intensity f at horizon 0. Negative weights are drawn as blue lines connecting neurons where orange lines show positive weights. The thicker the line, the higher the weight is in absolute value. Recall that each neuron is activated with a sigmoid function and the biases are not shown on the graph. In the input layer, all variables seem to be used in the computation of the first layer. In the first layer however, the second neuron presents a higher weight in the network than the others. In return, the inputs connected to the second neuron of the first hidden layer all present low relative weights. Even though the computational graph gives an overview of the neural network and is useful to understand the way the output is computed, it is not trivial to see which variables have more impact on the output. In the following section, I plot sensitivities of each variable in each horizon to better understand the causality of each input.

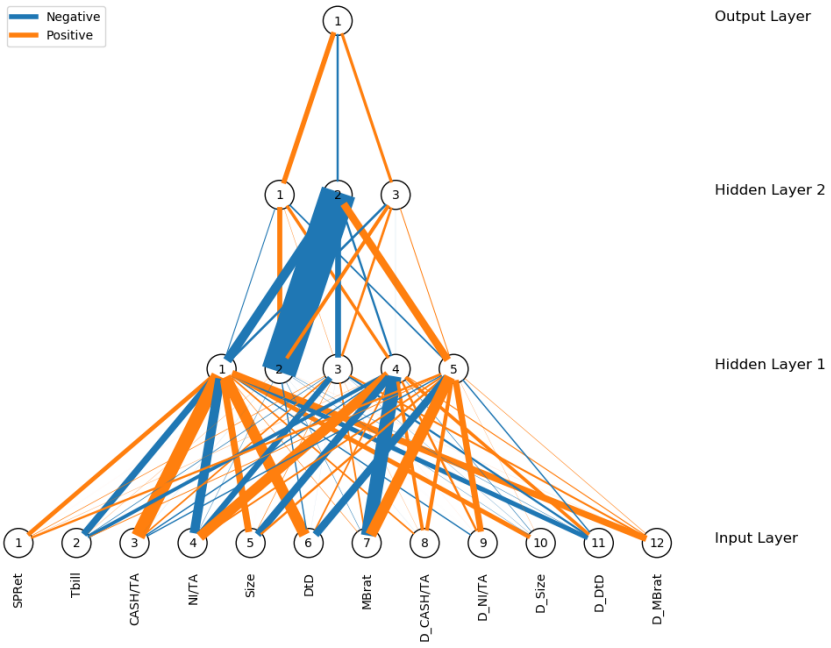


Figure 1.8: Trained [5, 3] Neural Network for forward default intensity at horizon 0
Negative weights are drawn as blue lines connecting neurons where orange lines show positive weights. The thicker the line, the higher the weight is in absolute value.

1.4.4 Sensitivities

Neural networks are often seen as black boxes because their outputs are coming from a general function involving many parameters. They are incorporating non-linearities via the layers and the activation functions. Figure 1.9 is an attempt at gauging how the model reacts to a change of an input variable. It plots the average default forward intensity against a shift in the specified variable and is computed the following way:

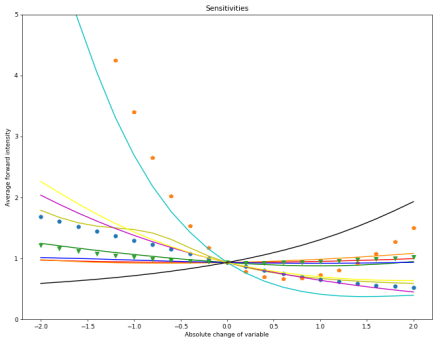
1. Compute the forward default intensity for all the observations in the test set and average the result. Keeping everything else equal, change the value of one feature by an absolute value and feed the updated observations to the network as new inputs and compute the new average forward default intensity.
2. Repeat step 1 for all absolute values in some interval.
3. Plot the average intensity against the absolute change.
4. Repeat steps 1-3 for all 11 other features.

As explained in Section 1.2.1, I impose a non-negativity constraint on forward intensities. A decreasing relationship means that an increase of the associated variable decreases the probability of default of the firm. A flat relationship means that the associated variable has a limited impact on the probability of default. We should expect CASH/TA, size, DtD, NI/TA, Market-to-book ratio and all their lagged differences (Δ) to be decreasing.

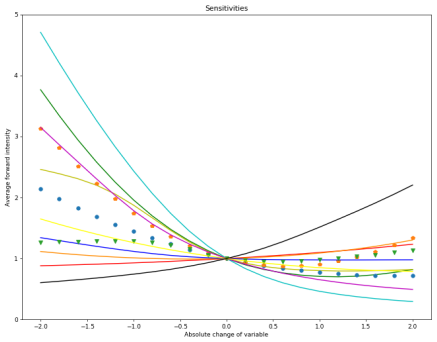
First of all, most graphs show non-linear relationships, which is not surprising given the nature of the neural network specification. Moreover, all relationships are intuitive and expected. Forward intensities in both Size and ΔSize are decreasing, suggesting that small firms tend to have a higher likelihood of defaulting, which is consistent with the “too big to fail" paradigm. Similarly, firms with decreasing cash (ΔCASH/TA < 0) or low levels of CASH/TA appear to have higher probabilities of default. The model also predicts that firms with low NI/TA or decreasing NI/TA should have higher probabilities of default. Finally, forward intensities in DtD should be decreasing for all horizons to reflect that a higher Distance-to-Default makes the firm less likely to default. At horizon 0, a negative change in Distance-to-Default has a substantially greater effect on forward intensities than any other variables. This result is consistent with Duffie et al. (2007). Overall, it appears that the most important predictors of default in this model are in the short term Market-to-book ratio, DtD and CASH/TA, and in the long run NI/TA, CASH/TA, ΔCASH/TA and DtD.

1.5 Conclusion

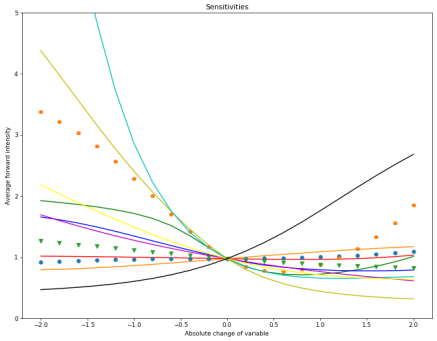
I propose an approach to estimate forward default intensities, which relies on using machine learning techniques. The key improvement over previous estimation methods is the introduction of possible highly non-linear relationships between covariates and forward intensities. Neural networks are nothing else than a very general mapping of input data to an output which is obtained by tuning weights while minimizing a given loss function. Non-linearities are introduced via the juxtaposition of layers and the activation functions. The econometric model governing the forward intensity written by Duan et al. (2012) has been adapted to this new framework to allow the use of neural networks. Neural networks are also well-suited for this chapter because they allow an easy replication of the benchmark model Duan et al. (2012). More specifically, if the network architecture is [1] (i.e. : one layer and one neuron) and the



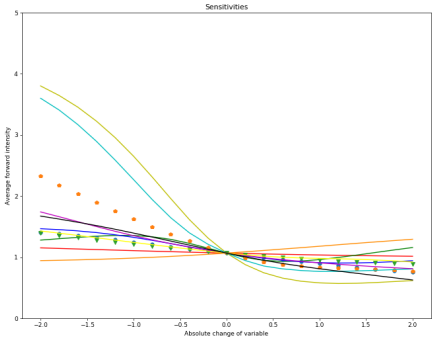
Horizon 0



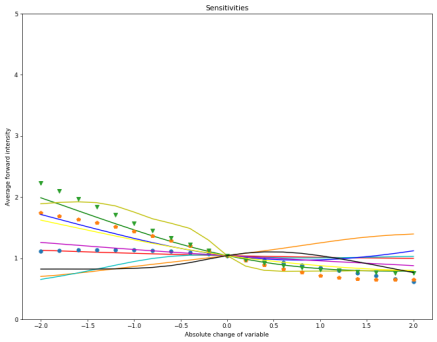
Horizon 3



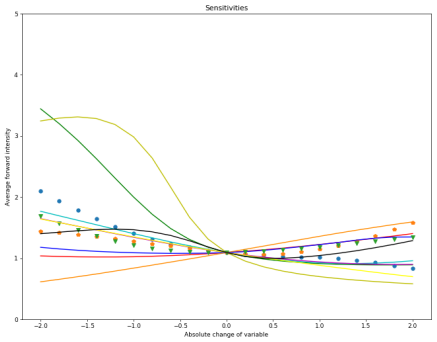
Horizon 6



Horizon 12



Horizon 24



Horizon 35

Figure 1.9: Sensitivities for each horizon
These plots show the average default forward intensity against a change in the specified variable, ceteris paribus.

activation function is chosen as being an exponential, the neural network boils down to a logistic regression. The dataset used in this chapter is the same as in previous literature to allow for an easier comparison. It consists of 5 firm-specific variables computed from accounting data and 2 macroeconomic variables to control for the health of the economy. I also account for momentum of these variables by feeding the model the one-lagged differences of each variable. Looking at summary statistics only, the average defaulted firm is small, has low cash, low market-to-book ratio, low Distance-to-Default and has large negative profits. To measure the discriminatory power of the models, I follow the previous literature and use Lorenz curves (also known as “cumulative accuracy profile" or “power curves"). The idea behind Lorenz curves is to order default probabilities and look at how they are distributed across defaulted and non-defaulted firms. The average Gini coefficient across all folds of the cross-validation is used as a comparison tool to gauge the accuracy of the model. The results show that the architecture [5, 3] (i.e. 2 layers with 5 neurons in the first hidden layer and 3 neurons in the second hidden layer) seems to outperform other architectures. In this setup, one can interpret the architecture as the non-linearity degree since a higher architecture involves more weights in the output function to be estimated. Out-of-sample Lorenz Curves and Gini coefficients show that the neural network approach outperforms the linear assumption for every horizon, suggesting the presence of non-linearities in forward default intensities. Finally, even if neural networks are known to be black boxes, I show how the model reacts to a change of input variables. Most sensitivities plots show non-linear relationships, which is not surprising given the nature of the neural network specification. It appears that the most important predictors of default in this model are in the short term Market-to-book ratio, Distance-to-Default and NI/TA, and in the long run NI/TA, CASH/TA, ΔCASH/TA and Distance-to-Default. Further works could involve more variables in the estimation. In particular, a challenging (due to lack of data) but nonetheless exciting study would be to explore the effect of market sentiment on default intensities.

2 Causal Networks with Neural Networks

2.1 Introduction

The causal network of a dynamical system provides important information that may help to better understand its behavior and ultimately design better policies to predict and control it. Large number of banks, insurances, hedge funds, and other financial institutions around the globe are interacting daily and thus their causal network is of great importance in econometrics.

There have been many attempts during the past decades to capture and visualize the network of interconnections among a set of financial institutions. A well-known time series causality measure used in econometrics is Granger-causality (Granger (1969)). This is based on statistical analysis of the financial time series such as their stock prices over a finite time period. Granger's definition of causality states that a time serie X is a cause of another time serie Y, if the one-step future forecast of Y is more precise when the forecasting information set includes X. Otherwise, when the forecast does not improve by including the information of X, then it is declared that X does not cause Y. This idea is reflected in the information-theoretic measure that we use in this chapter to infer the causal interactions among a network of time series.

Most empirical applications of Granger-causality have been studied with Vector Autoregressive (VAR) models. For instance, Billio et al. (2012) propose several measures based on Granger-causality to capture the interconnections between the returns of financial institutions on a monthly basis. It uses principle component analysis and pairwise Granger-causality tests to identify the causal networks. Other related works are Diebold and Yılmaz (2014) and Barigozzi and Hallin (2016) in which the authors propose connectedness measures based on generalized variance decomposition. However, the measures introduced in these works are again limited to linear systems and they are based on pairwise comparison which as we show in Section 2.2.2 fails to infer the true causal relationships.

Contributions of this chapter are both in network identification literature and in finance. Our contributions to network identification are as follows:

- We use directed information (DI), an information measure to infer the Granger-causalities among a set of time series. This measure is non-parametric (i.e. it does not depend on the underlying model of the dynamics) and it is capable of capturing causal relationships in both linear and nonlinear systems. The output of this approach is a directed graph known as Directed Information Graph (DIG) that visualizes the interconnections among a set of time series such as stock returns.
- Computing DI has high computational and sample complexity which makes it not suitable for inferring the causal structure of large networks. To solve this problem, we develop a novel approach based on Recurrent Neural Networks (RNNs) that reduces the complexity of evaluating DIs in high-dimensional settings.

Applications of DIG are various in finance. In particular, we recommend to use it as a preliminary feature selection of another predictive model. Feature selection is a process often used in machine learning and statistics which consists of keeping only a subset of relevant features, usually to avoid overfitting or to reduce dimensionality. For example, Piramuthu (2004) and Huang and Wang (2006) show that extraneous features are prone to reduce model's performance measures. Finance applications of feature selection models are various and include credit scoring, stock market behavior analysis or even fraud detection (Altinbas and Biskin (2015)). Tsai (2009) states that feature selection preprocessing is not addressed carefully enough in the bankruptcy prediction literature. They compare five feature selection methods used in bankruptcy prediction: step-wise regressions, correlation matrix, principal component analysis, t-tests and factor analysis and show that any of these methods improves performance. Yuqinq et al. (2013) uses a Sequential Forward Selection algorithm to select relevant features predicting the Turkish market index. The use of such feature selection model reduces model prediction error compared to the case where all features are used. This is due to information embedded in several economic factors already included in the market index.

2.1.1 Related Work

In recent years, several approaches have been developed to generalize the applicability of Granger-causality to non-linear and large dynamics. To mention a few, Psaradakis et al. (2005) introduce different terminologies for causality based on Granger's ideas and provide a set of parametric non-causality constraints in the context of Markov switching VAR models. In a similar context, Bianchi et al. (2019) investigate time-varying systemic risk based on a range of multi-factor asset pricing models and develop a Markov Chain Monte Carlo scheme to infer their model parameters and consequently obtain their corresponding networks. Another attempt is Bonaccolto et al. (2019) in which the authors explore quantile-based methods of Granger-causality. This method is consistent with Hong et al. (2009) and Corsi et al. (2018)

that focus on causality among tail events. These methods are suitable for capturing causal relationships that are not in the center of their distributions, or in the mean but in the tails of their distributions. It is important to emphasize that our proposed approach using DI is also capable of capturing such causal relations.

Most of the above aforementioned approaches are developed and tested for small size networks. Often, the problem of network identification in high dimensional settings requires more considerations and even its own techniques. For instance, Billio et al. (2019) propose a Bayesian non-parametric Lasso (BNPL) prior for high-dimensional vector autoregressive models that improve efficiency and accuracy. To overcome overfitting in large VAR models, BNPL clusters the vector autoregressive coefficients and shrinks the coefficients of each cluster toward a common location. However, this method is limited to linear models with Gaussian innovations. To overcome this limitation, Kalli and Griffin (2018) propose a Bayesian non-parametric approach that allows for nonlinearity in the conditional mean, heteroskedasticity in the conditional variance, and non-Gaussian innovations. However, unlike the BNP-Lasso, it does not allow sparsity in the model. Petrova (2019) proposes yet another non-parametric, quasi-Bayesian likelihood estimation methodology for high dimensional setting with time-varying parameters. The work in Iacopini and Rossini (2019) tackles the curse of dimensionality by a two-stage approach. First, a spike-and-slab prior distribution is used for each entry of the coefficient matrix which also identifies the interconnection network. In the second stage, it imposes prior dependence on the coefficients by specifying a Markov process for their random distribution. A closely related work is Bernardi and Costola (2019) that proposes a shrinkage and selection methodology designed for network inference in high-dimensional settings. It uses a regularized linear regression model with spike-and-slab prior on the parameters. However, both methods are limited to VAR models.

The remainder of this chapter is strutured as follows. Section 2.2 reviews the notion of Granger-causality and formally introduces directed information graphs which are suitable for linear and nonlinear systems. Section 2.3 introduces a novel approach for inferring the Granger-causal network of high dimensional nonlinear systems. Section 2.4 applies our methodology to learn the causal network of both synthetic and real-world dataset. For the real-world experiment, we use the daily stock prices of US firms.

2.2 Causal Network

We present in this section our econometric approach to learn the causal dependencies in a dynamical systems based on Granger-causality (see Granger (1969)). We begin by introducing some notations. Plain capital letters denote random variables or processes, while lowercase letters denote their realizations. Bold letters are used for column vectors, matrices, and tensors and calligraphy letters are used for sets. We use $X_{j,t}$ to denote the value of a time series X_j at time t and X_j^t to denote the time series X_j up to time t. For a set $\mathscr{A} = \{a_1, ..., a_n\}$ and an index set $\mathscr{I} \subseteq \{1, ..., n\}$, we define $\mathscr{A}_{-\mathscr{I}} := \mathscr{A} \setminus \{a_i : i \in \mathscr{I}\}$.

2.2.1 Granger Causality

Various frameworks and graphical models have been developed to capture and represent interconnections among variables or processes. One of the most popular and widely used frameworks in economics is the notion of Granger-causality originally introduced by Wiener (1956) and generalized by Granger (1969). We say that X Granger-causes Y if the prediction of the future of Y is more precise using all available information (including X) than using all available information except X.

This formulation was originally implemented using multivariate autoregressive (MVAR) models and linear regressions. In particular, let $\{X, Y, Z\}$ be three time series. In order to identify the influence of X_t on Y_t, Granger's idea is to compare the performance of two linear regressions[1]: the first one predicts Y_t given $\{X^{t-1}, Y^{t-1}, Z^{t-1}\}$ and the second one predicts Y_t given $\{Y^{t-1}, Z^{t-1}\}$. If they perform similarly, then we say X does not Granger-cause Y.

To go beyond linear systems, works such as Quinn et al. (2015) and Massey (1990) use information-theoretical measures and generalize Granger-causality. In this chapter, we introduce and apply directed information (see Quinn et al. (2015)), an information-theoretical tool to measure interconnections among firms. Directed information (DI) has been used in many applications to infer causal relationships. For example, Quinn et al. (2011) and Kim et al. (2011) use it for analyzing neuroscience data and Etesami and Kiyavash (2014) and Etesami et al. (2018) apply it to market data.

Directed information graphs (DIGs) have been developed to visualize the inferred interconnections among time series (see Quinn et al. (2015)). DIGs are a type of graphical models in which nodes represent time series and arrows indicate the direction of causation. We use DIG to represent the causal network among the covered firms.

2.2.2 Directed Information Graphs (DIGs)

We describe in this section how the directed information is able to capture the connections in causal networks. Next, we formally define directed information graphs (DIG).

We define a dynamical system constituted of three time series $\{X, Y, Z\}$ that we assume have a joint probability density function $p(X, Y, Z)$. Granger-causality states that to know whether X influences Y or not, we need to compare the performance of two predictors of Y. The first predictor uses the history of all information available (i.e. $\{X, Y, Z\}$) while the second predictor uses only the history of $\{Y, Z\}$ If the former performs better than the latter, X has information on Y. However, if they perform equally, it is an indication that X is not causing Y in this time interval. To rigorously formalize this idea, we need the predictors and a measure to compare their performances.

[1] Note that this formulation is only applicable in linear systems.

In the definition of DI, the predictors belong to the space of probability measures. More precisely, the prediction of the first predictor at time t is $p(Y_t|Y^{t-1}, Z^{t-1}, X^{t-1})$ that is the conditional density function of Y_t given the history of all time series. Similarly, the prediction of the second predictor is $p(Y_t|Y^{t-1}, Z^{t-1})$ that is the conditional density function of Y_t given the history of all time series except time series X.

Given the predictions of the first and the second predictors at time t for an outcome $y_t \in \mathcal{Y}$, the goodness of these predictions are measured by the log-loss that are defined respectively by

$$-\log p(Y_t = y_t|Y^{t-1}, Z^{t-1}, X^{t-1}),$$
$$-\log p(Y_t = y_t|Y^{t-1}, Z^{t-1}).$$

According to the above measures of goodness, the better the predictor is, the smaller its log-loss will be. This loss function also has meaningful information-theoretical interpretations. Namely, the log-loss is the Shannon's code length[2], i.e., the number of bits required to efficiently represent y_t (see Etesami et al. (2018)).

At time t for an outcome $y_t \in \mathcal{Y}$, the difference between the log-losses of the two predictors compares their performances. This difference is also called regret, denoted r_t :

$$\begin{aligned} r_t :&= -\log p(Y_t = y_t|Y^{t-1}, Z^{t-1}) - \big(-\log p(Y_t = y_t|Y^{t-1}, Z^{t-1}, X^{t-1})\big) \\ &= \log \frac{p(Y_t = y_t|Y^{t-1}, Z^{t-1}, X^{t-1})}{p(Y_t = y_t|Y^{t-1}, Z^{t-1})}. \end{aligned} \tag{2.1}$$

Regrets are always positive for all t and all outcomes y_t. Over the time interval $[1, T]$, the average regret is

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[r_t], \tag{2.2}$$

where the expectation is taken over the joint density function[3] of X, Y, and Z, i.e.,

$$\mathbb{E}[r_t] = \int p(y^t, z^{t-1}, x^{t-1}) \log \frac{p(y_t|y^{t-1}, z^{t-1}, x^{t-1})}{p(y_t|y^{t-1}, z^{t-1})} dy^t dx^{t-1} dz^{t-1}. \tag{2.3}$$

The average regret in (2.2) is the directed information (DI). We use it as the measure of causation in this chapter. This measure is always positive and if it is zero, it is an indication that the history of the time serie X does not contain significant information helping in the prediction of the time serie Y given the past of the time series Y and Z. We can generalize this definition to more than three time series as follows,

[2] It is also called the description length of y_t. For more information see Cover and Thomas (2012).
[3] For the sake of notational simplicity, we use $p(y^t, z^{t-1}, x^{t-1})$ to denote $p(Y^t = y^t, Z^{t-1} = z^{t-1}, X^{t-1} = x^{t-1})$.

Definition 2.2.1. Consider a network of m time series $\mathscr{R} = \{R_1, ..., R_m\}$ with the joint probability density function p. The directed information from R_i to R_j over the time interval $[1, T]$ is given by

$$I(R_i \to R_j || \mathscr{R}_{-\{i,j\}}) := \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\log \frac{p(R_{j,t} | \mathscr{R}_{-\{i,j\}}^{t-1}, R_j^{t-1}, R_i^{t-1})}{p(R_{j,t} | \mathscr{R}_{-\{i,j\}}^{t-1}, R_j^{t-1})} \right], \tag{2.4}$$

where $\mathscr{R}_{-\{i,j\}}^{t-1} := \{R_1^{t-1}, ..., R_m^{t-1}\} \setminus \{R_i^{t-1}, R_j^{t-1}\}$. We say that R_i has influence on R_j over the time interval $[1, T]$, if and only if

$$I(R_i \to R_j || \mathscr{R}_{-\{i,j\}}) > 0. \tag{2.5}$$

An interpretation of R_i influencing R_j is that varying R_i will change the value of R_j even if all the other variables within the network remains unchanged. In another words, if R_i does not influence R_j, then varying R_i would not change R_j when the values of the remaining times series are fixed. This can be seen from the fact that DI compares two conditional distributions of R_j over a time horizon of length T; one is conditioned on the past of all time series while the other one is conditioned on all the history except R_i. Thus, if DI in (2.4) is zero, then these two conditional distributions are equal over this time horizon. This implies that the history of R_i does not contain any useful information for R_j.

Note that the definition of DI does not rely on any model assumption, thus DI is capable of inferring the causal relationships in general (linear or non-linear) dynamical systems. Next, we define the graphical model that we use in this chapter to visualize the causal network among firms.

Definition 2.2.2. We denote by DIG the directed information graph of a set of m time series $\mathscr{R} = \{R_1, ..., R_m\}$ which consists of a directed graph $G = (\mathscr{V}, \mathscr{E})$, where $\mathscr{V}$ stands for the set of nodes representing time series $\mathscr{R}$ while $\mathscr{E}$ stands for the set of arrows indicating influences between time series (i.e. an arrow from R_i to R_j indicates that R_i has an influence on R_j).

A simple way to represent the DIG G of a dynamical system is via the adjacency matrix $\mathbf{DIG} = [d_{i,j}]_{m \times m}$ that is defined by

$$d_{j,i} = \begin{cases} 1 & \text{if } I(R_i \to R_j || \mathscr{R}_{-\{i,j\}}) > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

Given a DIG $G = (\mathscr{V}, \mathscr{E})$, we define the parent set of node R_j denoted by $\mathscr{PA}_j \subset \mathscr{V}$ to be the set of all times series that have direct influences on R_j, i.e., $\mathscr{PA}_j := \{R_k : d_{j,k} = 1\}$. Similarly, the children set of node R_j is given by $\mathscr{CH}_j := \{R_k : d_{k,j} = 1\}$. Example 1 demonstrates the DIG of a simple linear system.

Example 1. *Let $\{X, Y, Z\}$ be a network of three time series with the following dynamics,*

$$\begin{pmatrix} X_t \\ Y_t \\ Z_t \end{pmatrix} = \begin{pmatrix} 0.5 & 0 & 0 \\ 0.4 & 0.5 & 0 \\ 0 & -0.2 & 0 \end{pmatrix} \begin{pmatrix} X_{t-1} \\ Y_{t-1} \\ Z_{t-1} \end{pmatrix} + \begin{pmatrix} N_{X_t} \\ N_{Y_t} \\ N_{Z_t} \end{pmatrix}, \tag{2.7}$$

where N_X, N_Y, and N_Z are three independent stationary Gaussian processes with zero mean and a diagonal covariance matrix (1, 0.9, 1). Since the dynamics is linear and the exogenous noises are Gaussian, we can compute the DIs using the following expression[4] (see Etesami and Kiyavash (2014)).

$$I(Z \to Y||X) = \frac{1}{2T} \sum_{t=1}^{T} \log \frac{|\Sigma_{Y_{t-1},Y_t,X_{t-1}}||\Sigma_{Z_{t-1},Y_{t-1},X_{t-1}}|}{|\Sigma_{Y_{t-1},X_{t-1}}||\Sigma_{Z_{t-1},Y_{t-1},Y_t,X_{t-1}}|}, \tag{2.8}$$

where $|\Sigma_{Y_{t-1},Y_t,X_{t-1}}|$ denotes the determinant of the covariance matrix of $\{Y_{t-1}, Y_t, X_{t-1}\}$. Using (2.8)*, we compute the DIs of this system,*

$$\begin{aligned} &I(Y \to X||Z) = 0, \;\; I(Z \to X||Y) = 0, \;\; I(Z \to Y||X) = 0, \;\; I(Z \to Z||X, Y) = 0, \\ &I(X \to Z||Y) = 0, \quad I(X \to Y||Z) \approx 0.1, \quad I(Y \to Z||X) \approx 0.03. \end{aligned} \tag{2.9}$$

Figure 2.1 illustrates the DIG of the system (2.7)*. In this example, $\mathscr{PA}_Z = \{Y\}$ and $\mathscr{CH}_Z = \{\}$.*

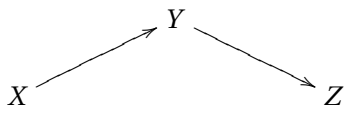


Figure 2.1: DIG estimated from the directed informations (2.9)
X influences Y, Y influences Z. This DIG correctly estimates the dynamics of the system in (2.7).

Inference methods based on pairwise comparison has been developed and applied in the literature to identify the causal structure of time series. The methods in Billio et al. (2012), Billio et al. (2010), and Allen et al. (2010) are three such examples. However, pairwise comparison is not a correct approach in general and may fail to capture the true underlying network. For instance, considering the pairwise comparison in Example 1 between X and Z leads to a conclusion that X directly influences Z, which would be inaccurate. More precisely, without conditioning on Y, we obtain

$$I(X \to Z) = \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\log \frac{p(Z_t|Z^{t-1}, X^{t-1})}{p(Z_t|Z^{t-1})}\right] \approx 0.002 > 0.$$

Notice that the DI in (2.4) is not a measure based on pairwise comparison. On the contrary, it measures the influence by conditioning on the remaining time series within the network.

[4] Equation (2.8) does not hold in more general settings.

2.2.3 Inferring DIGs

Inferring the DIG of a dynamical system requires estimating the DIs between all ordered pairs of time series within that system. More precisely, inferring the DIG of a network of m time series requires computing $m(m-1)$ number of DIs. On the other hand, estimating DI requires estimating all the expectation terms in (2.4). In information theory this expectation is known as *conditional mutual information*[5], i.e.,

$$I(R_{j,t}; R_i^{t-1} | \mathscr{R}_{-\{i,j\}}^{t-1}, R_j^{t-1}) := \mathbb{E}\left[\log \frac{p(R_{j,t} | \mathscr{R}_{-\{i,j\}}^{t-1}, R_j^{t-1}, R_i^{t-1})}{p(R_{j,t} | \mathscr{R}_{-\{i,j\}}^{t-1}, R_j^{t-1})}\right]. \tag{2.10}$$

Using this notation, (2.4) can be written as follows

$$I(R_i \to R_j || \mathscr{R}_{-\{i,j\}}) = \frac{1}{T} \sum_{t=1}^{T} I(R_{j,t}; R_i^{t-1} | \mathscr{R}_{-\{i,j\}}^{t-1}, R_j^{t-1}), \tag{2.11}$$

Therefore, parametric and non-parametric estimators for the conditional mutual information can be used to estimate the DIs. Given i.i.d. samples of the time series, it exists two main methods to estimate the terms in (2.11) : the plug-in empirical estimator and the k-nearest neighbor estimator. For an overview of such estimators, we refer to the articles in Paninski (2003), Noshad et al. (2019), Sricharan et al. (2011) and Jiao et al. (2013).

In general, estimating the DI in (2.11) has high sample complexity because it requires estimating high dimensional conditional distributions. However, information about the underlying dynamics can simplify the learning task of the DIG. For instance, in Example 1, since the underlying dynamics is linear with Gaussian exogenous noises, the DIs can be computed via the covariance matrices (2.8). Clearly, the covariance matrix can be estimated with lower complexity compared to conditional mutual information. For our experimental results, we used (2.8) for the linear Gaussian experiment and the k-nearest neighbor estimator in for the non-linear experiment. The main reason for selecting k-nearest method is because it usually shows better performance in comparison to the other estimators.

Side information can also help to infer the DIG of a dynamical system without directly estimating the DIs but instead providing an alternative approach to identify the DIG. For example, if it is given that the underlying dynamics is linear, i.e., $\mathbf{X}_t = \mathbf{A}\mathbf{X}_{t-1} + \mathbf{N}_t$, then it has been shown in Etesami and Kiyavash (2014) that the support[6] of the coefficient matrix $\mathbf{A}$ is equal to the adjacency matrix of its corresponding DIG. This result implies that in linear systems, one can obtain the DIG by estimating the coefficient matrix. The latter problem has lower complexity and it can be done using e.g., linear regression. For similar examples in econometric models see Etesami et al. (2018).

[5] See Cover and Thomas (2012) for more details.

[6] The support of a matrix $\mathbf{B} = [b_{i,j}]$ is a binary matrix of the same dimension as $\mathbf{B}$ such that its entry (i, j) is one if and only if $b_{i,j} \neq 0$.

2.2.4 DIG in High-dimensional Settings

For large networks with thousands nodes and millions of edges such as social or financial networks, DIGs become too complex to infer and analyze. The main reason is that without any side information, estimating the DI has high computational and sample complexity. Furthermore, the estimating complexity of DI increases with the dimension of the network. This is due to the fact that the DI in (2.4) measures the influence from R_i to R_j by conditioning on the information from the remaining network $\mathscr{R}_{-\{i,j\}}$. Therefore, the size of the conditioning set grows with the size of the network. This motivates the prior works to reduce the complexity of estimating DIs and thus make it more suitable for inferring the DIG of large networks by reducing the size of the conditioning set.

One such approach is proposed by Quinn et al. (2013), in which they developed an efficient algorithm to identify the best directed tree approximation of a given network. This means reducing the size of the conditioning set to zero, i.e., no conditioning. However, this approach comes with the price of an approximation error and furthermore it fails to identify many interconnections between the processes.

The authors in Quinn et al. (2017) present a more generalized version of the above approximation in which they identify the optimal connected bounded in-degree[7] approximations. This method reduces the size of the conditioning set in (2.4) to some constant value (bound of the in-degrees) which is independent of the network size. Although, this approach improves upon the approximation error but there is still a trade-off between the sample complexity and the approximation error. In another words, as the in-degree bound increases, the sample complexity increases but the approximation error decreases.

In this chapter, we propose a new method that reduces the size of the conditioning set in (2.4) to only one for any given network while introducing less approximation error compared to the prior works. In this method, we estimate the directed information from R_i to R_j by conditioning on an auxiliary time series. This auxiliary time series is defined such that it comprises the information that the remaining of the network $\mathscr{R}_{-\{i,j\}}$ has about R_j. Next section explains this idea in more details.

[7] Connected bounded in-degree graphs with bound k are connected directed graphs in which each node has at most k number of parents.

2.3 Methodology

In order to present our method, we need the following preliminary result that characterizes an important property of DI in (2.4). All the proofs are presented in the Appendix A.2.

Lemma 1. *Consider a network of m time series $\mathscr{R} = \{R_1, ..., R_m\}$ with corresponding DIG $G = (\mathscr{V}, \mathscr{E})$. Let $\mathscr{C}$ be a subset of $\mathscr{R}_{-\{i,j\}}$ such that $\mathscr{PA}_j \subseteq \mathscr{C}$. If $R_i \notin \mathscr{PA}_j$, then we have*

$$I(R_i \to R_j || \mathscr{C}) = 0. \tag{2.12}$$

Note that if $\mathscr{C} = \mathscr{R}_{-\{i,j\}}$ and R_i is not a parent of R_j, then by the definition of DIG, Equation (2.12) holds. On the other hand, this result states that to detect whether there is an influence from R_i to R_j in a network of time series, it suffices to find a subset of time series that either contains the parents of R_j or their information. In the remaining of this section, we first clarify the above statement via a simple linear system and later generalize it to non-linear models using neural networks.

Remark 1. *It is important to emphasize that the reverse of Lemma 1 does not hold. In another words, if there exists a subset $\mathscr{C} \subset \mathscr{R}_{-\{i,j\}}$ such that* (2.12) *holds, we cannot conclude that R_i has no direct influence on R_j.*

2.3.1 Linear Systems

Consider a first order vector autoregression model (VAR) with m time series,

$$\mathbf{X}_t = \mathbf{A}\mathbf{X}_{t-1} + \mathbf{N}_t, \tag{2.13}$$

where $\mathbf{X}_t, \mathbf{N}_t \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{m \times m}$, and $\mathbf{N}_t$ is a vector of m independent exogenous noises. As we discussed earlier, the result in Etesami and Kiyavash (2014) implies that the DIG of this VAR model is encoded in the support of its coefficient matrix $\mathbf{A} = [a_{i,j}]$, i.e.,

$$I(X_i \to X_j || \mathscr{X}_{-\{i,j\}}) = 0 \Longleftrightarrow a_{j,i} = 0. \tag{2.14}$$

In another words, the parents of time series X_j are the ones whose corresponding coefficients are non-zero in the j-th row of matrix $\mathbf{A}$. This also can be seen from the j-th row of the matrix equation in (2.13),

$$X_{j,t} = \sum_{k=1}^{m} a_{j,k} X_{k,t-1} + N_{j,t}. \tag{2.15}$$

Another way to interpret the above equation is to say that the information of the network about time series X_j is in the form of a “portfolio", i.e., a linear combination of the other time series. Therefore, it is possible to summarize the network's information about X_j into only one time series, namely a well-designed portfolio. Next result shows the form of such portfolio.

Lemma 2. *In the linear system of* (2.13)*, X_i has no direct influence on X_j if and only if*

$$I(X_i \to X_j || Q) = 0, \tag{2.16}$$

where Q is a time series which we call the ideal portfolio and it is defined by $Q_{t-1} := \boldsymbol{u}_t^T \boldsymbol{X}_{-\{i\},t-1}$, where

$$\boldsymbol{u}_t := \arg\min_{\boldsymbol{w}\in\mathbb{R}^{m-1}} \mathbb{E}\left[||X_{j,t} - \boldsymbol{w}^T \boldsymbol{X}_{-\{i\},t-1}||_2^2\right],$$
$$\boldsymbol{X}_{-\{i\},t-1} := [X_{1,t-1}, \cdots, X_{i-1,t-1}, X_{i+1,t-1}, \cdots, X_{m,t-1}]^T.$$

According to the above Lemma, projecting X_j on $\mathscr{X}_{-\{i\}}$ results in an ideal portfolio Q that contains all the information for deciding whether there is an influence from X_i to X_j. Hence, instead of estimating $I(X_i \to X_j || \mathscr{X}_{-\{i,j\}})$ whose complexity depends on the network size, one can estimate $I(X_i \to X_j || Q)$. Note that the sample complexity of the latter DI does not grow with the size of the network and thus it is suitable for estimating the DIG of large networks.

2.3.2 Non-linear Systems with Additive Noise

Inferring the causal network of non-linear systems is a challenging problem that its complexity increases exponentially with the dimension of the network. In this section, we study the causal inference problem in non-linear systems whose dynamics can be captured by

$$X_{j,t} = F_j(\mathscr{X}^{t-1}) + \varepsilon_{j,t}, \quad j = 1, ..., m, \tag{2.17}$$

where $\mathscr{X}^{t-1} = \{X_1^{t-1}, ..., X_m^{t-1}\}$, $\{F_j(\cdot)\}$ is a set of non-linear continuous functions, and $\{\varepsilon_{j,t}\}$ is a set of independent exogenous noises. We call this model non-linear with *additive noise* due to the noise term that is added to the non-linear term[8]. This is a general non-linear dynamics that can be used to model the behavior of wide range of physical dynamical systems. The dynamics is called *Markovian* if $\mathscr{X}^{t-1}$ is replaced by $\mathscr{X}_{t-1} = \{X_{1,t-1}, ..., X_{m,t-1}\}$.

Below, we generalize the result of Lemma 2 to the non-linear system in (2.17) by showing that in such systems, it is possible to reduce the conditioning set in the DI to one time series.

[8] In contrary to additive noise, there are systems in which the exogenous terms are multiplicative, e.g., $X_{j,t} = X_{i,t-1}\varepsilon_{j,t} + X_{j,t-1}$.

Lemma 3. *In* (2.17)*, X_i has no direct influence on X_j if and only if*

$$I(X_i \to X_j || Q) = 0, \tag{2.18}$$

where Q is a time series defined by $Q_{t-1} := F_j(\mathscr{X}^{t-1}_{-\{i\}})$.

In the remaining of this section, we propose two methods to obtain the time series Q introduced in the above Lemma.

Koopman-based lifting technique

Consider a particular sub-class of the non-linear system in (2.17) whose dynamics is defined by

$$F_j(\mathscr{X}^{t-1}) = \sum_{k=1}^{K} w_{j,k} h_k(\mathscr{X}_{t-1}), \;\; j = 1, ..., m, \tag{2.19}$$

where $\{w_{j,k} \in \mathbb{R}\}$ are the weights and $\{h_k(\cdot)\}$ denotes a set of library functions that are assumed to be known. This model is Markovian and the library functions can be seen as a set of basis that are used to approximated $F_j(\cdot)$. Examples of such library functions are monomials and Gaussian radial basis functions.

In this setting, the results of Lemma 3 implies that the following time series can be substituted in the conditioning of the DI.

$$Q_t = \sum_{k=1}^{K} w_{j,k} h_k(\mathscr{X}_{-\{i\},t-1}). \tag{2.20}$$

However, in this formulation, the weights $\{w_{j,k}\}$ are unknown. An approach to obtain the weights is a non-linear filtering technique known as Koopman-based lifting (see Koopman (1931)). This technique takes observational data and a set of library functions as inputs and obtains the unknown coefficients $\{w_{j,k}\}$. The main steps of this technique are transforming the data (lifting the data), applying a linear identification on the lifted data, and finally applying another transformation to bring down the results into the original vector field. Figure 2.2 illustrates the main steps. For more details see Appendix A.2 and Mauroy and Goncalves (2019).

Although, the Koopman-based lifting technique is theoretically sound but it has some shortcomings facing real-world applications. First, the Koopman's performance depends on the choice of the library functions and second, it often fails to estimate the real time series Q. More precisely, this technique involves the computation of matrix $\mathbf{L} := log(\mathbf{P}_x^{\dagger}\mathbf{P}_y)/T_s$, where $\mathbf{P}_x$ and $\mathbf{P}_y$ are estimated from the observational data[9]. Matrix $\mathbf{P}_x^{\dagger}$ denotes the pseudo-inverse, and the function $\log(\cdot)$ denotes the (principal) matrix logarithm. On the other hand, Koopman

[9] See Appendix A.2 for more details.

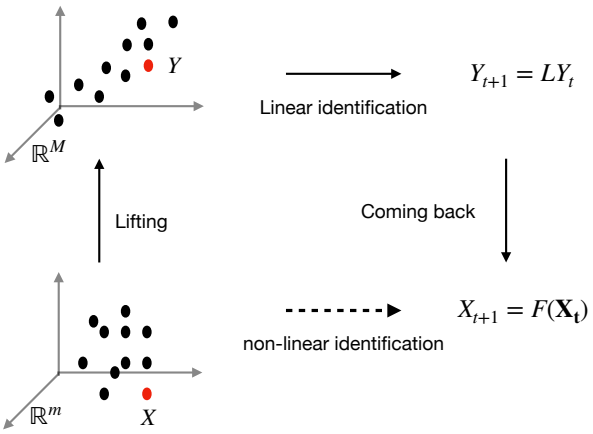


Figure 2.2: Koopman lifting technique compared to classical non-linear identification
Starting from the bottom left plot, the main steps of this technique are transforming the data (lifting the data), applying a linear identification on the lifted data, and finally applying another transformation to bring down the results into the original vector field.

lifting technique is applicable for estimating the time series Q only when the resulting matrix $\mathbf{L}$ is real[10]. However, this is not always the case in real-world applications due to observational noises and lack of sufficient data. To overcome such shortcomings, we propose an alternative approach to estimate Q using recurrent neural networks (RNNs).

RNNs method

Recurrent neural networks are a specific class of neural networks well suited to learn time series. They are distinguished by their memory as they are able to remember information from prior inputs to influence their current outputs. The universal approximation theorem states that a neural network with enough hidden layers can approximate any non-linear continuous function such as $F_j(\cdot)$ in (2.17) (see Hornik et al. (1989)).

Given the aforementioned result, we train a RNN with LSTM layers using the observational data to estimate the time series Q defined in Lemma 3. More precisely, our RNN maps $\mathscr{X}_{-\{i\}}^{t-1}$ as the inputs to $X_{j,t}$ as the output. We choose the architecture of the network to be pyramid-like. That is, the width of layer k is strictly bigger than the width of layer $k+1$. The pyramidal structure is known to retain and improve accuracy while reducing the number of parameters (see Ullah and Petrosino (2016) and Tripathy and Bilionis (2018)). In our case, the first hidden layer has 100 units, the second hidden layer has 50 units, the third hidden layer has 40 units and the fourth hidden layer has 30 units. In every layer, the activation function that we use for the recurrent step (i.e. : forget, input and output gates) is the sigmoid function while the activation function for the cell and hidden states is the hyperbolic tangent. Finally, to avoid overfitting and to allow the network to generalize better on out-of-sample data, we use dropout on every layer. Dropout is a regularization technique often used in neural networks where

[10] See Culver (1966) for conditions under which a real matrix has a real logarithm.

connections within the LSTM network are randomly selected and excluded from updates in the training process. This has the effect of introducing noise in the training process because every training step is performed with a different network layout and it allows more nodes to be involved. In our case, the fraction of units that we drop (i.e : the dropout rate) in each layer is 0.5, 0.3, 0.2, 0.1, respectively.

Let $R_j(\mathscr{X}_{-\{i\}}^{t-1};\Theta^*)$ denotes the trained RNN with parameters Θ^*. In this case, the time series Q can be written as $Q_{t-1} = R_j(\mathscr{X}_{-\{i\}}^{t-1};\Theta^*)$. Finally, we use (2.18) to detect whether X_i has influence on X_j or not. Algorithm 1 summarizes the steps of our RNN method.

Algorithm 1: Infer-DIG

Input: Observational data of m time series up to time T, $\mathscr{X}^T$, Threshold $\alpha > 0$;
Output: Adjacency matrix of **DIG** = $[d_{i,j}]$;
for $i, j = 1, ..., m$ **do**
  Train a RNN $R_j(\cdot;\Theta^*)$ that maps $\mathscr{X}_{-\{i\}}^{t-1}$ to $X_{j,t}$;
  Define $Q_{t-1} = R_j(\mathscr{X}_{-\{i\}}^{t-1};\Theta^*)$;
  if $I(X_i \to X_j||Q) > \alpha$ **then**
    $d_{j,i} = 1$
  else
    $d_{j,i} = 0$

2.4 Experimental Results

Since the true empirical DIG of firms is unknown, to evaluate the performance of our approach, we use different simulated environment. In this section, we first describe the simulation methodology in a linear Gaussian framework. We then show that our results generalize well to nonlinear setting by conducting an experiment on a nonlinear system. Finally, we apply our approach to a set of empirical data describing the daily stock prices of US firms and obtain their corresponding causal network.

2.4.1 Linear Gaussian Framework

In this experiment, we consider a linear system, a VAR(1) model whose dynamics are given by

$$\mathbf{X}_t = \mathbf{A}\mathbf{X}_{t-1} + \mathbf{N}_t \tag{2.21}$$

with m being the number of asset returns, $\mathbf{X}_t = (X_{1,t}, X_{2,t}, ..., X_{m,t})^\top$ being the vector of returns at time t, $\mathbf{A} = [a_{i,j}]$ being a $m \times m$ matrix and $\mathbf{N}_t$ being a $\mathscr{N}(0,\mathbf{I})$ vector of noises. As we discussed earlier, in such linear systems, $a_{i,j}$ captures the influence of asset j on asset i, i.e., there is an influence from j to i if and only if $a_{i,j} \neq 0$.

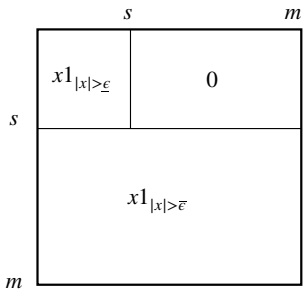


Figure 2.3: Adjency matrix A for the linear Gaussian framework
Structure of matrix A in (2.21) that is built using (2.22).

To reflect an important property of the market that some firms are more connected than others in our experiment, we divided the m time series into two parts. First part ($1 \le i \le s$) indicates assets with high degrees of connectedness and the second part ($1+s \le i \le m$) are the ones with low degrees of connectedness. Parameter $1 < s < m$ denotes the numbers of assets with high degrees of connectedness. Afterward, for every entry (i, j) of $\mathbf{A}$, we independently generated a random number $x \sim U(-0.9, 0.9)$ and decided on value $a_{i,j}$ as follows,

$$a_{i,j} = \begin{cases} x1_{|x|>\underline{\epsilon}}, & \text{if } 1 \le i \le s, 1 \le j \le s, \\ x1_{|x|>\bar{\epsilon}}, & \text{if } 1+s \le i \le m, 1 \le j \le m, \\ 0, & \text{if } 1 \le i \le s, 1+s \le j \le m, \end{cases} \tag{2.22}$$

where $1_{a>b}$ denotes the indicator function which is equal to 1 when $a > b$ and 0 otherwise and $\underline{\epsilon}$ and $\bar{\epsilon}$ are thresholds to define non-zero entries in the upper-left and the lower part of $\mathbf{A}$, respectively.

Figure 2.3 illustrates the structure of the resulting $\mathbf{A}$. We select these thresholds such that $\underline{\epsilon} < \bar{\epsilon}$. This ensures that the upper-left of $\mathbf{A}$ is denser than its lower part or equivalently, assets with indices $\{1, ..., s\}$ are more connected than the ones with indices $\{1+s, ..., m\}$. In our experiment, we select $(s, m) = (85, 100)$ and $(\underline{\epsilon}, \bar{\epsilon}) = (0.4, 0.7)$. Finally, to guarantee the stability of the time series, we rescale[11] $\mathbf{A}$ such that its spectral radius is strictly less than one, i.e., $\rho(\mathbf{A}) < 1$. Once the matrix $\mathbf{A}$ is defined, we simulate the time series using (2.21) for a period of $T = 30000$ and use the resulting data for our estimations.

To study the effect of the conditioning set on detecting the influences, in our experiments, we consider four different conditioning sets. More precisely, to measure whether asset i influences asset j, we estimate $I(X_i \to X_j || \mathscr{C}_j)$ for the following choices of the conditioning set:

[11] Formally, we use $\mathbf{A}/(\rho(\mathbf{A}) + \epsilon)$, where $0 < \epsilon$.

1. *True parents:* In this approach, we select $\mathscr{C}_j$ to be the true parents of X_j excluding X_i, i.e., $\mathscr{C}_j = \mathscr{PA}_j \setminus \{X_i\}$. Note that this approach is not practical[12] and we use it only as the benchmark to better understand the performances of the other approaches.

2. *Most correlated:* In this case, we define $\mathscr{C}_j$ to be the set of k most correlated assets with X_j (except X_i).

3. *Ideal portfolio:* In this scenario, $\mathscr{C}_j$ contains the portfolio Q, where Q is defined in Lemma 2. For further discussion see Appendix A.2.

4. *RNN:* This method applies Algorithm 1 to estimate the time series Q and defines $\mathscr{C}_j = \{Q\}$.

Note that we also applied the Koopman-based lifting techniques but due to its mentioned shortcomings, it was unable to robustly identify the interconnections. Hence, we could not compare its performance with the other methods. In this experiment, since the dynamics is linear and the noises are Gaussian, we use Equation (2.8) to estimate the DIs. Finally, we obtain the adjacency matrix of the corresponding DIGs by comparing the estimated DIs with a threshold $\alpha > 0$, i.e.,

$$[\mathbf{DIG}]_{j,i} = \begin{cases} 1 & \text{if } \widehat{I}(X_i \to X_j || \mathscr{C}_j) > \alpha, \\ 0, & \text{otherwise,} \end{cases} \tag{2.23}$$

where $\widehat{I}(X_i \to X_j || \mathscr{C}_j)$ denotes the estimated DI from X_i to X_j given the conditioning set $\mathscr{C}_j$. In order to compare the performances of the aforementioned four approaches, we use the precision and recall measures between the true DIG (obtained from $\mathbf{A}$) and their estimated DIGs. Formally, the precision and the recall are defined by

$$Precision := \frac{TP}{TP+FP}, \quad Recall := \frac{TP}{TP+FN},$$

where

$$TP := \sum_{i,j=1}^{m} 1_{a_{j,i}\neq 0} 1_{[\mathbf{DIG}]_{j,i}\neq 0}, \quad FP := \sum_{i,j=1}^{m} 1_{a_{j,i}=0} 1_{[\mathbf{DIG}]_{j,i}\neq 0},$$
$$FN := \sum_{i,j=1}^{m} 1_{a_{j,i}\neq 0} 1_{[\mathbf{DIG}]_{j,i}=0}.$$

Figure 2.4 shows the performances of the four aforementioned approaches in the linear framework. It is not surprising that the *true parents* approach achieves 100% accuracy, as it is anticipated by Lemma 1. The *ideal portfolio*'s performance is guaranteed by Lemma 2 and it is verified by our experiment. However, it is important to emphasize that the *ideal portfolio*

[12] This is because in structural learning problems, we do not know the true parents of each asset. In another words, if we had access to the true parents of each asset, we would have the DIG of the system and there is no need to compute the DIs.

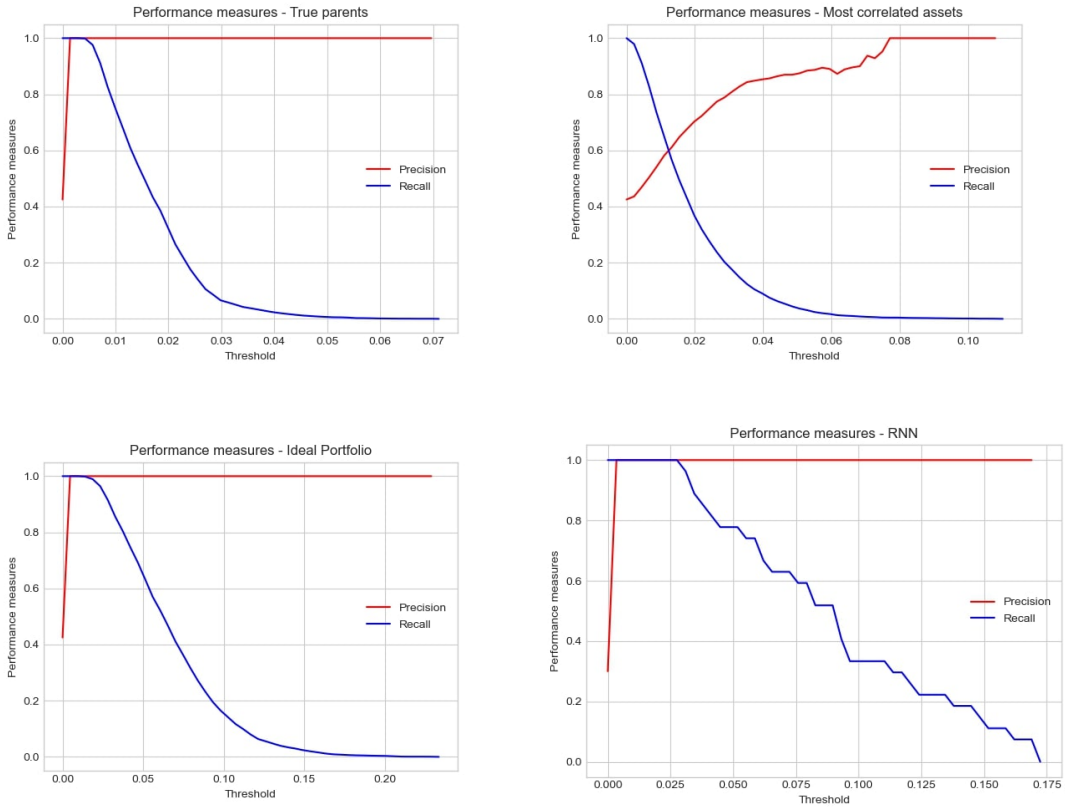


Figure 2.4: Precision and recall curves in the linear framework.
Precision and recall curves for the *true parents*, *most correlated*, *ideal portfolio*, and the *RNN*, respectively.

shows ideal performance because the underlying model is linear. As we will see in the next section, its performance declines when the underlying model deviates from being linear. For the *most correlated* approach, we used $k = 10$ but as it is shown in Figure 2.4, it has the worst performance among the four conditioning methods. This is due to the fact that the set of the ten most correlated assets with a given asset j does not necessarily contains the true parents of asset j. On the other hand, we observe high accuracy from the *RNN* approach which is a striking result. This result is an evidence that a RNN is capable of estimating the ideal portfolio, i.e., the time series Q in Lemma 2 without any side information about the underlying model.

2.4.2 Non-Linear framework

To compare the performances of the different approaches from the previous section in a non-linear environment, we simulate a set of quadratic processes whose dynamics is given below,

$$X_{i,t} = b_i \mathbf{X}_{t-1}^T \mathbf{A}_i \mathbf{X}_{t-1} + N_{i,t},\ i = 1, ..., m, \tag{2.24}$$

where $\mathbf{A}_i \in \mathbb{R}^{m\times m}$, $\mathbf{X}_t = (X_{1,t}, X_{2,t}, ..., X_{m,t})^T$, $N_{i,t} \sim \mathcal{N}(0,\sigma^2)$, and $b_i \sim U(-0.9, 0.9)$. Note that the term $|[\mathbf{A}_i]_{j,k} + [\mathbf{A}_i]_{k,j}|$ captures the effect of $X_{j,t-1}X_{k,t-1}$ on $X_{i,t}$. Thus, it is possible to obtain the true parents of asset i as follows,

$$\mathscr{PA}_i = \{X_j : [\mathbf{1}^T \cdot \left(|\mathbf{A}_i^T + \mathbf{A}_i|\right)]_j > 0\}, \tag{2.25}$$

where $\mathbf{1}$ denotes all-one vector of length m. Each matrix $\mathbf{A}_i$ is simulated independently by following the similar procedure as in Section 2.4.1. In this experiment, since the model is non-linear, we could not apply (2.8) to estimate the DIs but instead we used the k-nearest method to estimate the mutual information and applied Equation (2.11).

Herein, we again compare the performances of the four different conditioning approaches. Figure 2.5 shows the precision-recall curves for these approaches in the quadratic model with $m = 15$. Precision-recall curves are a standard tools to illustrate and compare the performances of different learning methods. In this curve the precision is demonstrated in the y-axis vs. the recall on the x-axis for all potential values of the threshold α.

Similar to the linear setting, we use the *true parents* as a benchmark since it has the ideal performance. It is however important to emphasize that this conditioning approach has higher complexity compared to the others. This is because in the *true parent* approach, the size of the conditioning set is relatively larger than the other approaches.

For the *most correlated* approach, we use $k = 5$, i.e., the size of the conditioning set is five. With this method, we could slightly reduce the estimation complexity of the DIs compared to the *true parent* approach but this comes with the price of losing the performance. Clearly, the performance of the *most correlated* approach can be improved by increasing k but this will increase the complexity.

The performance of the *ideal portfolio* approach (using the time series in Lemma 2 as the conditioning) is worse than all others which is not surprising as the model is no longer linear. This means that the information embedded in the linear portfolio is not sufficient to decide the non-linear influences among the time series.

Finally, as it is shown in Figure 2.5, the *RNN* approach outperforms the *most correlated* and the *ideal portfolio* approaches and it shows close performance to the *true parents* but with the size of the conditioning set equal to one. This result once more fortifies our claim that with a RNN we can summarize the information of the network into one time series and use it for detecting the causal relationships. This claim is due to Lemma 3 and the universal approximation theorem which states that a neural network with enough hidden layers can approximate any non-linear function (see Hornik et al. (1989)). The slight difference between the performance of the *RNN* and the *true parents* is because of the estimation error in the recurrent neural network.

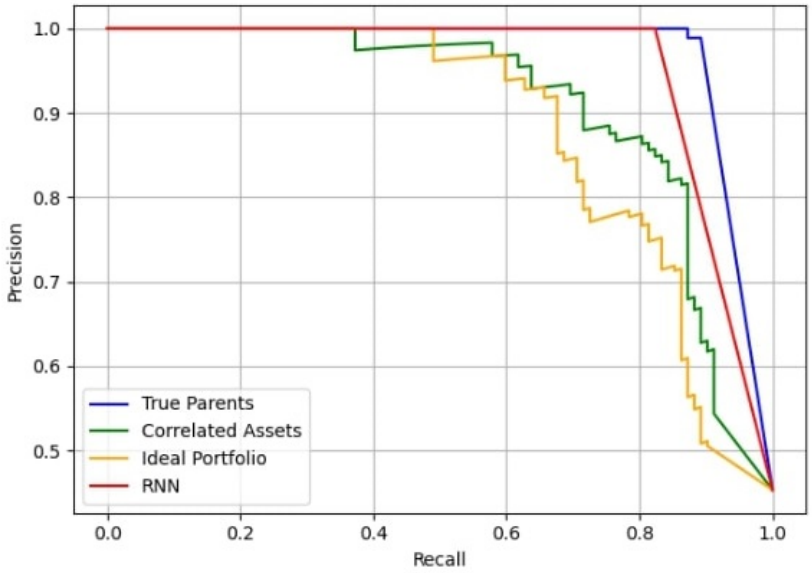


Figure 2.5: Precision-recall curves for the quadratic model
Blue, green, yellow and red lines show the precision-recall curve of the *true parents*, *most correlated*, *ideal portfolio*, and the *RNN*, respectively.

2.4.3 Empirical DIG

This section describes how to apply our approach to empirical data and obtain the DIG of US firms. We extracted the daily stock prices and the daily US Treasury rate as risk free returns from the CRSP database from 1990 to 2020. As the market is likely to evolve through these years, we chose to divide the dataset into six subsets, each of which has a length of five years and estimate the corresponding DIG of each subset separately. Herein, we assume that the causal structure of the market evolved but its rate was slow enough such that during a period of five years, the DIG of the market remained unchanged.

For every subset, we keep the data of the 1000 firms with the highest maximum market capitalization and compute their excess return time series $X_{i,t}$, using the following relationship,

$$X_{i,t} = ln(P_{i,t}) - ln(P_{i,t-1}) - r_t, \tag{2.26}$$

where $P_{i,t}$ denotes the stock price of the firm i at time t and r_t is the risk free rate at time t. Afterwards, we apply Algorithm 1 with the excess returns as the input to estimate the corresponding DIG of each subset. We use the k-nearest neighbor method to estimate the DIs. We define the threshold α to be the unconditional mean across the estimated DIs. Note that in this experiment, the true DIGs of the market are not known, hence, we could not compute the precision-recall curves.

1990-1994	1995-1999	2000-2004	2005-2009	2010-2014	2015-2019
0.23	0.27	0.18	0.28	0.32	0.25

Table 2.1: Degree of Granger Causality (DGC) for each sub-graph.
DGC is defined as the fraction of relationships in the network among all potential relationships.

For the sake of presentation, instead of the complete DIGs with 1000 nodes, we draw the sub-graphs consisting of the 30 largest firms in Figures[13] 2.6, 2.7, and 2.8 . Each graph consists of 60 nodes illustrating the cause firm on the top hemisphere and the effect firm on the bottom hemisphere. For instance, if there is an edge between "from: AAPL" on the top and "to: GOOGL" on the bottom, it means that Apple influences Google. The dynamic evolution of the DIGs through time can often be explained by real events that happened in the market. For instance, in the DIG 2010-2014, Apple was not influencing General Electric (GE). However, on the 17th October 2017, Apple announced a partnership with GE to bring Predix, GE's data and analytics platform, to their iPhones and iPads. We are able to capture this partnership in the DIG 2015-2019 as an edge is now present from Apple to GE. Another example is the announced collaboration between AT&T and Cisco to manage IoT devices and launch 5G service at the end of the 2010s: there was neither an edge from AT&T to Cisco nor from Cisco to AT&T during the first half of the 2010s, but the DIG for the second half of the 2010s shows a mutual influence, reflecting an increased relationship between the two companies.

Table 2.1 shows the Degree of Granger-Causality (DGC) defined as the fraction of relationships in the network among all potential relationships. Formally,

$$DGC = \frac{1}{N^2} \sum_i \sum_j [\mathbf{DIG}]_{j,i}, \tag{2.27}$$

These results show that the DGC increased both in the DotCom bubble and in the Subprime Crisis, suggesting an increase of the connectedness in turmoil periods. This finding is consistent with Longin and Solnik (2001) stating that correlation increases in bear markets.

Tables 2.2 and 2.3 show the outdegree and indegree of every firm in the six subsets. Outdegree is defined as the number of edges going out of a specific node. Indegree is the number of edges going to a specific node. These tables also reveal interesting facts. For instance, the SPY ticker, an ETF launched in 1993 and aiming at tracking the S&P500 return, enters in the 30 biggest market capitalizations in 2010 and has the highest number of outdegrees in the periods 2010-2014 and 2015-2019 but relatively low number of indegrees. This result suggests that the market return is influencing a high number of firms, but the converse is not necessarily true.

[13] For a better presentation, interactive plots are available at https://marcaureledivernois.github.io/firm-network/

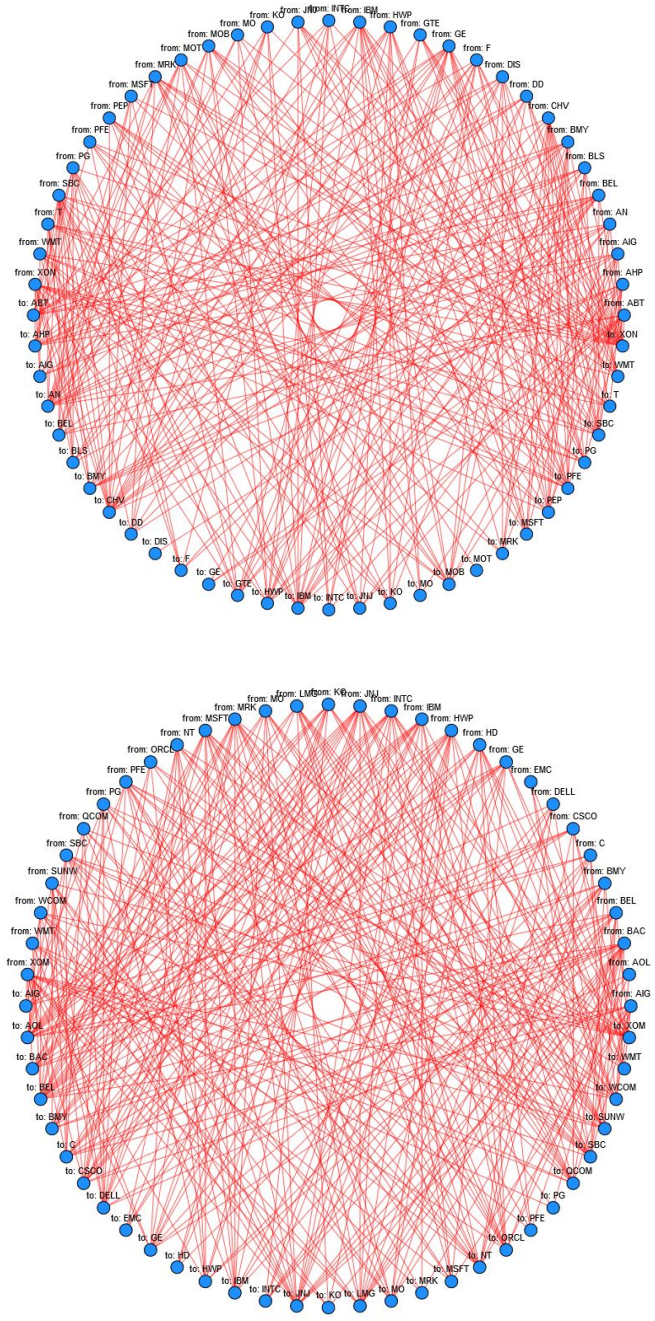


Figure 2.6: Empirical DIG for the periods 1990-1994 and 1995-1999.
Empirical DIG for the periods 1990-1994 (top) and 1995-1999 (bottom). Interactive graphs can be found at https://marcaureledivernois.github.io/firm-network/

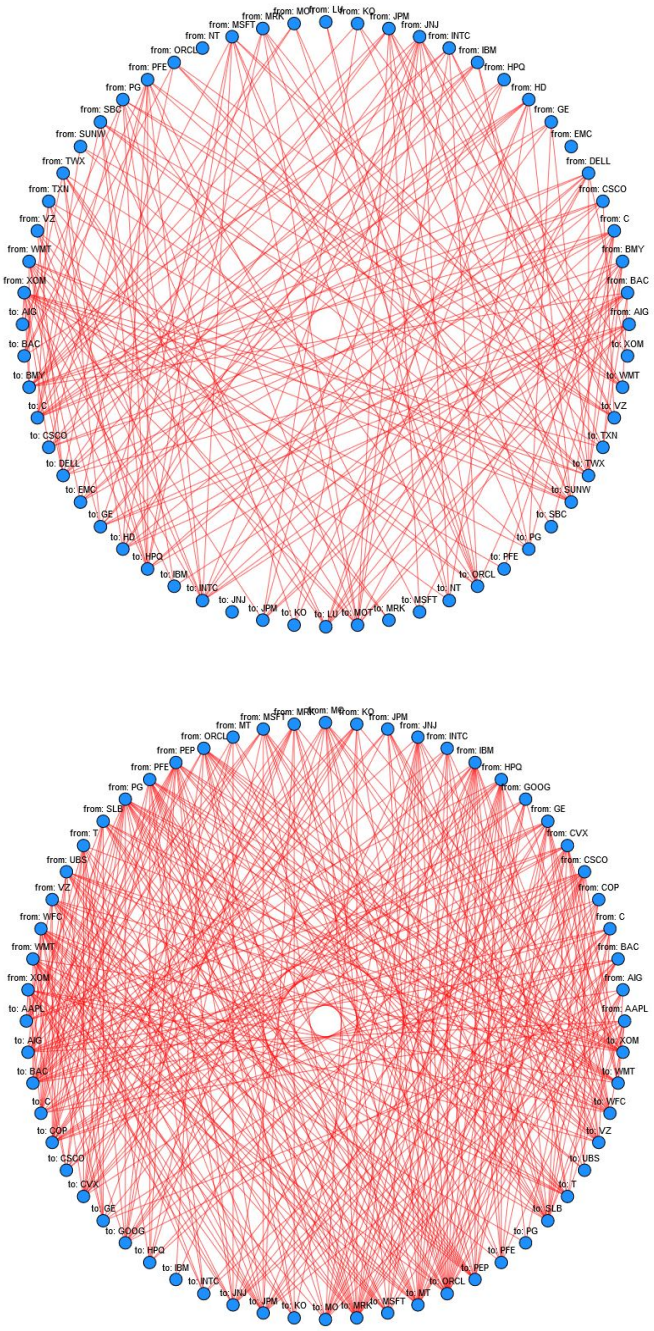


Figure 2.7: Empirical DIG for the periods 2000-2004 and 2005-2009.
Empirical DIG for the periods 2000-2004 (top) and 2005-2009 (bottom). Interactive graphs can be found at https://marcaureledivernois.github.io/firm-network/

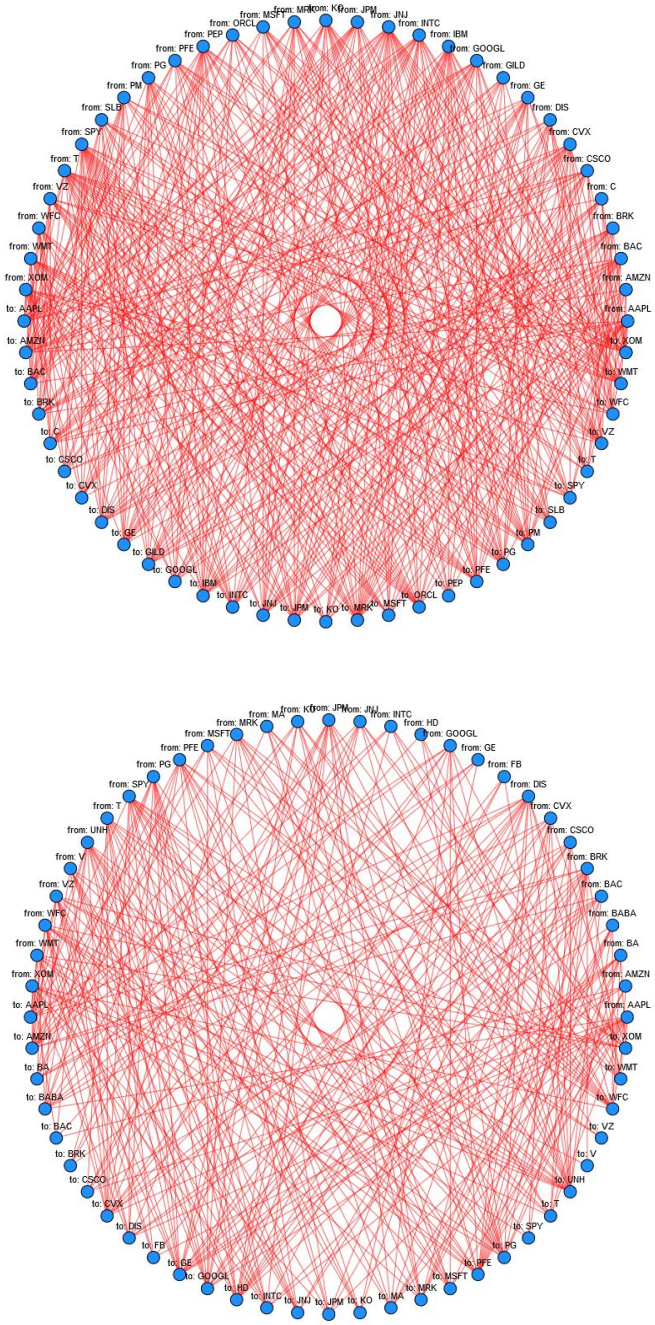


Figure 2.8: Empirical DIG for the periods 2010-2014 and 2015-2019.
Empirical DIG for the periods 2010-2014 (top) and 2015-2019 (bottom). Interactive graphs can be found at https://marcaureledivernois.github.io/firm-network/

1990-1994		1995-1999		2000-2004		2005-2009		2010-2014		2015-2019	
Ticker	Out	Ticker	Out	Ticker	Out	Ticker	Out	Ticker	Out	Ticker	Out
SBC	13	JNJ	12	XOM	12	PEP	15	SPY	17	SPY	14
CHV	11	HWP	12	JNJ	10	WFC	13	T	15	DIS	12
XON	10	INTC	12	MSFT	9	CSCO	12	JNJ	15	BRK	12
PG	10	BAC	12	DELL	9	WMT	11	CVX	13	PFE	11
GE	10	PFE	11	WMT	9	IBM	11	AAPL	12	UNH	11
AN	9	IBM	11	C	9	PG	11	INTC	12	JPM	11
BEL	9	MSFT	11	HD	8	CVX	11	CSCO	12	AAPL	10
MRK	8	SBC	11	JPM	8	HPQ	10	GE	12	GOOGL	10
IBM	8	GE	10	PFE	8	VZ	10	IBM	11	WMT	9
BMY	8	MRK	10	PG	7	GE	9	GOOGL	11	CSCO	9
ABT	7	LMG	9	INTC	7	UBS	9	XOM	11	VZ	9
AHP	7	MO	9	BAC	7	PFE	9	JPM	10	BA	9
T	7	NT	9	BMY	6	C	9	KO	10	CVX	8
BLS	7	XOM	9	SBC	6	SLB	9	MRK	10	WFC	8
AIG	7	KO	9	IBM	5	MSFT	9	BRK	10	V	8
F	7	HD	8	TWX	5	ORCL	9	SLB	10	PG	8
HWP	7	BEL	8	AIG	5	MRK	8	WMT	10	T	7
MOT	7	BMY	8	CSCO	4	KO	8	VZ	9	JNJ	7
DD	6	SUNW	8	GE	4	GOOG	7	DIS	9	MRK	7
DIS	6	AIG	7	VZ	4	XOM	7	PFE	9	KO	6
MOB	6	WCOM	7	MRK	4	JNJ	7	BAC	8	XOM	5
PEP	6	CSCO	6	MOT	4	MO	7	PM	8	MSFT	5
MSFT	6	QCOM	6	TXN	4	JPM	6	ORCL	7	MA	5
INTC	6	C	6	HPQ	3	COP	6	PG	7	BABA	5
WMT	6	PG	6	ORCL	3	T	6	PEP	6	FB	4
GTE	5	DELL	5	KO	2	INTC	6	MSFT	6	BAC	4
MO	5	ORCL	4	SUNW	1	AAPL	6	C	5	INTC	4
JNJ	4	WMT	4	LU	1	BAC	5	WFC	5	GE	3
PFE	4	EMC	2	EMC	0	MT	4	AMZN	5	HD	3
KO	2	AOL	2	NT	0	AIG	4	GILD	5	AMZN	2

Table 2.2: Outdegrees ranked for each sub-graph.
Outdegree (Out) is defined as the number of edges going out of a specific node.

1990-1994		1995-1999		2000-2004		2005-2009		2010-2014		2015-2019	
Tic	In	Tic	In	Tic	In	Tic	In	Tic	In	Tic	In
XON	22	BEL	15	MOT	12	MRK	19	MRK	18	UNH	17
CHV	17	JNJ	15	INTC	11	ORCL	19	PM	15	PFE	16
AN	13	AOL	14	C	11	BAC	16	JPM	15	HD	14
SBC	12	XOM	14	SUNW	11	SLB	14	ORCL	14	AMZN	14
HWP	11	NT	12	ORCL	10	WFC	13	IBM	14	GOOGL	12
PEP	10	QCOM	12	BMY	9	JPM	12	XOM	13	GE	12
AHP	10	SUNW	11	DELL	9	COP	12	PG	13	PG	12
MOB	10	ORCL	11	TWX	7	MT	11	JNJ	12	CVX	11
GTE	9	CSCO	11	HPQ	7	PEP	10	DIS	12	BABA	9
IBM	9	C	11	GE	7	C	10	INTC	11	MRK	9
BMY	8	DELL	10	CSCO	7	MO	10	WFC	10	DIS	9
ABT	8	SBC	10	LU	6	CVX	10	C	10	WFC	8
KO	7	BMY	9	HD	6	AIG	9	VZ	10	BA	7
BEL	7	WCOM	9	TXN	6	T	9	WMT	10	MA	7
MSFT	7	GE	9	EMC	6	JNJ	8	GE	10	AAPL	7
BLS	6	IBM	9	BAC	5	GOOG	7	KO	9	T	7
PFE	6	MO	9	PG	5	KO	7	BAC	9	WMT	6
DD	6	LMG	8	WMT	5	VZ	6	PFE	9	MSFT	6
JNJ	5	HWP	6	VZ	5	MSFT	6	AAPL	9	INTC	6
WMT	4	MSFT	6	JPM	4	INTC	6	SPY	8	FB	5
F	4	BAC	5	KO	3	AAPL	6	CVX	7	KO	5
MRK	4	INTC	5	NT	3	GE	6	AMZN	7	JPM	4
PG	4	WMT	5	XOM	2	WMT	6	GILD	7	XOM	4
AIG	4	PFE	4	MSFT	2	XOM	5	CSCO	7	JNJ	4
T	3	AIG	4	AIG	2	PFE	5	GOOGL	7	SPY	4
GE	2	KO	3	MRK	1	CSCO	4	MSFT	6	BRK	4
MO	2	MRK	2	PFE	1	HPQ	4	PEP	5	CSCO	3
INTC	2	PG	2	SBC	1	UBS	2	SLB	5	BAC	2
DIS	1	HD	2	IBM	0	IBM	1	T	4	V	1
MOT	1	EMC	1	JNJ	0	PG	1	BRK	4	VZ	1

Table 2.3: Indegrees ranked for each sub-graph.
Indegree (In) is defined as the number of edges going to a specific node.

2.5 Conclusion

In this chapter, we introduce an information-theoretic measure known as directed information that is capable of capturing nonlinear Granger-causality in an interactive system. We develop a novel algorithm based on recurrent neural network utilized with directed information. This algorithm can infer the interconnections within a large network with less complexity than previous works. As a proof of concept, we show that our approach performs well both in a linear and in a non-linear simulated environments. Finally, we apply this algorithm to infer the causal relationships among the major US firms during 1990 to 2020.

3 StockTwits Classified Sentiment and Stock Returns

3.1 Introduction

Can the stock market be predicted by analyzing social media? Recent developments in machine learning and the growing quantities of available text data from online news, social media and annual reports have triggered intensive research in finance. In their pioneering paper, Antweiler and Frank (2004) compute a bullishness measure out of 1.5 million messages posted on Yahoo! Finance and Raging Bull and find that stock messages help predict market volatility. Their results clearly reject the hypothesis that all that talk is just noise. They show that there is financially relevant information present in social media. In a similar vein, Tetlock (2007) constructs a measure of media pessimism from a Wall Street Journal column and finds that it predicts downward pressure on market prices.

Most of the previous financial studies of social media rely on pre-defined or manually annotated sentiment dictionaries. Such approaches are limited in various ways. How to create a sentiment classifier that understands the vocabulary of the messages posted by the investors? For instance, “bull" is an animal in everyday language but refers to someone optimistic in the financial jargon. Loughran and McDonald (2011b) create a words list, which helps classify tone in a financial document. However, this might not be sufficient in the context of social media because messages posted present many typos, abbreviations and slang, so one needs to have an additional layer of data preprocessing. For instance, the word “goooooood" would not be recognized by the model if it is not corrected into “good" first. On the other hand, manually annotating and validating dictionaries is not a scalable approach to handling social media content.

This chapter overcomes these limitations. We develop a machine learning algorithm to classify the sentiment of a large sample of StockTwits messages as bullish, bearish, or neutral. The sample consists of all messages referring to US and Canadian stocks, including ETFs and other types of securities available on CRSP/Compustat, from January 2010 to March 2020. We train our machine learning classifier on the set of all user sentiment-labeled messages, which constitute about one third of the sample. We then classify the sentiment of all remaining

messages. Our method scales and performs very well. It achieves an out-of-sample accuracy of 85.9%, which compares well to the anecdotal 80 to 85% probability that human annotators agree on the sentiment of a document (see, e.g., Wilson et al. (2005) and Chen et al. (2020)). As a side product, we generate a vocabulary of one million investor sentiment-labeled terms consisting of up to three words that frequently appear in StockTwits messages.

We then construct a stock-aggregate daily sentiment polarity measure and relate it to daily stock returns. We find that polarity is positively associated with contemporaneous returns. However, unconditionally, polarity cannot predict next-day returns, which is in line with the efficient market hypothesis (EMH). We then conduct an event study. We define events as days of sudden peaks of message volume of individual tickers. We classify events as bullish, bearish, or neutral depending on the prevailing polarities. We find that bullish (bearish) events are strongly associated with large positive (negative) abnormal returns. Cumulative abnormal returns over the preceding 20 days of an event have no predictive power on the type of event. Returns normalize immediately after the jump on the event date, which again is in line with the EMH. In contrast, remarkably, we find that cumulative abnormal polarity has statistically significant predictive power on the type of event. We assess the economic relevance of our findings with the performance of cumulative abnormal polarity ranked portfolios. We find that for appropriate choices of thresholds, cumulative abnormal polarities provide valuable signals for stock market investments.

As a technical byproduct, we develop a sentiment classifier of micro-blogs for imbalanced data. This addresses the stylized fact that bloggers post more bullish than bearish-labeled messages. In our sample, the ratio is five to one. On the other hand, we find that not all messages carry a substantial stock market relevant sentiment. Rather than re-sampling from the underrepresented bearish class, we thus introduce an auxiliary neutral class. We then run two independent binary classifiers. The first (second) classifies messages as bullish versus non-bullish (bearish versus non-bearish). We aggregate the two binary outcomes and classify a message as bullish (bearish) for the concordant combination bullish/non-bearish (non-bullish/bearish), and neutral otherwise. This approach is very simple and efficient, and eliminates the class imbalance bias at the same time. It builds on any traditional binary classifier. We use logistic regression on Term Frequency-Inverse Document Frequency (TFIDF)-vectorized messages. TFIDF is a weighting scheme gauging the importance of a word in a document.

This chapter contributes to the growing literature on machine learning classification of social media and its interaction with the stock market. Most previous financial studies use Twitter as their primary source of data. Twitter has the advantage of being used by a wide range of people across the world and a few influencers can attract the attention of many investors. In 2013, following a meeting with Tim Cook (Apple CEO), Carl Icahn tweeted that he bought a large position in Apple and believed that the company is extremely undervalued. This bullish tweet caused the market capitalization of Apple to jump by $12 billion. In 2019, JPMorgan has created the Volfefe Index to track Donald Trump's tweets impact on the stock market.

However, it is more difficult to disentangle noise from relevant tweets in Twitter than in other more focused social media. Results from Ghoshal and Roberts (2016) show that StockTwits is significantly more informative than Twitter data. This is not surprising as StockTwits is a finance-focused platform whereas Twitter also captures irrelevant opinions on a wide range of non-finance related matters.

This chapter is the first work that analyzes the predictive power of StockTwits messages on stock returns unconditionally and around specific events. Renault (2017) builds an intraday investor sentiment indicator using messages and finds that the change in investor sentiment of the first half-hour of a trading day helps forecast the last half-hour market return of that trading day. However, his classifier is based on a dictionary consisting of 8 thousand manually validated and modified terms, which limits its scalability. Renault (2020) uses larger data sets and compares various classifiers, including machine learning.

Our approach is in some parts similar to Ranco et al. (2015), who also study the relation of micro-blog sentiments with stock returns. However, they use Twitter data, whereas the finance-tailored StockTwits data we use results in higher contemporaneous correlations between stock returns and polarity. They manually annotate 100 thousand tweets, which limits the scalability of their approach. Our sample is much larger (90 million versus 1 million messages) and covers a longer period (10 years versus 13 months).

Ke et al. (2020) extract sentiment from news articles on the Dow Jones Newswires. They train a sentiment score directly on returns. In contrast, we use user sentiment-labeled StockTwits messages as the training and validation set for our sentiment classifier.

This chapter also contributes to the EMH literature by gauging how cumulative average abnormal returns and abnormal polarities behave around sudden peaks of message activity.

The remainder of the chapter is structured as follows. Section 3.2 discusses StockTwits and stock market data. Section 3.3 develops our sentiment classifier based on TFIDF vectorization. Section 3.4 introduces the sentiment polarity measure and relates it to stock returns. Section 3.5 contains the event study. Section 3.6 discusses the sentiment-sorted portfolio performance. Section 3.7 concludes. The appendix contains additional statistics and background material.

3.2 StockTwits and Stock Market Data

StockTwits is a large social media platform similar to Twitter but designed for investors and traders. Users register online and can post messages about any listed stock through the prefix $ followed by the ticker of the stock. StockTwits was created in 2008 as an app built on the Twitter's API and later detached from Twitter to build a standalone social network. As of April 2019, it had over two million registered users and the number of daily posted messages has been growing exponentially, see Figure 3.1.

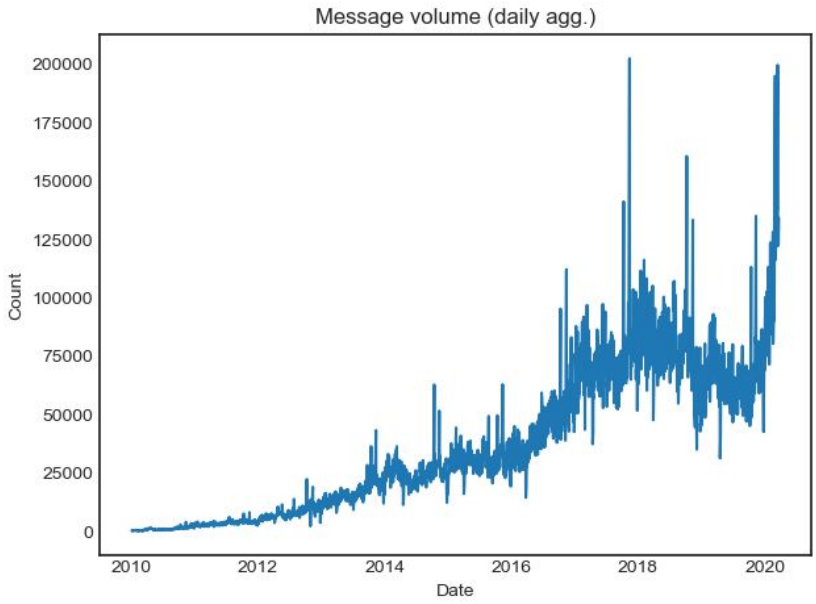


Figure 3.1: Number of messages posted daily on StockTwits
Message volume posted on StockTwits. Numbers are aggregated daily.

StockTwits describes itself as "the voice of social finance and the best way to find out what is happening right now in the markets and stocks you care about". In practice, it is effectively used by finance professionals to express their opinions on individual stocks and the market as a whole. Importantly, users have the option to label their posted messages as either bullish or bearish.[1] This feature is key for our approach, as it allows for sentiment classification of all messages using machine learning trained on the user-labeled messages.[2]

The reasons for using StockTwits and not other social media data for financial studies are at least threefold. First, a major challenge in applying natural language processing is the creation of an appropriate labeled vocabulary. Loughran and McDonald (2011a) show that it is essential to have a specific vocabulary to interpret finance documents (i.e., many words have a different meaning in finance than in traditional English, such as "bear trap"). In addition to that, social media slang is an additional layer of language complexity. To this extent, the functionality to self-tag bullish and bearish messages on StockTwits is extremely valuable as it allows the creation of a specific labeled vocabulary out of labeled messages. We are not aware of any other social media platform in finance offering this functionality. Second, text data from StockTwits is more reliable and less noisy than from general purpose platforms, such as Twitter, because messages focus on finance and economics matters only. Micro-bloggers have incentives to post valuable information in order to maintain or increase mentions and retweets, and thus have a greater share of voice in the forum (Sprenger et al. (2014)). On the other hand, StockTwits messages might be biased and subject to malicious users that try to

[1] This optional label was effectively available as of mid-2010.
[2] One third of the messages in our sample have a user labeled sentiment.

manipulate the market. However, market manipulations likely happen only rarely as the SEC closely monitors potential influencers to prevent any market abuse. Third, extracting data from StockTwits is easy because of its API. StockTwits' API is designed to query the database to download messages via JSON requests. We provide a short tutorial in Appendix A.3.1.

We use stock market data from CRSP/Compustat. We extract daily closing prices, daily volume of transactions and number of shares outstanding for all US and Canadian stocks, as well as ETFs and some other types of securities, from January 2010 to March 2020. Stock prices and number of shares are adjusted to account for any distribution (i.e., dividends, stock splits) so that a comparison can be made on an equivalent basis before and after the distribution. We use as risk-free rate the 3-month US T-bill rate, converted into daily risk-free returns. We henceforth refer to daily stock excess returns over risk-free simply as returns. Using a Python script, we then extract all messages from StockTwits for the list of tickers corresponding to the sample of US and Canadian stocks. This results in 90 million messages, which we download and store as JSON files.[3] Overall, our sample covers 8843 tickers, whereof 75% refer to ordinary common share, 15% to ETFs, and the remaining 10% to other types of securities. Henceforth, we interchangeably refer to any of these securities as either a stock or a ticker.

Every StockTwits message includes eight features: (1) the reference ticker(s), (2) a timestamp, (3) a unique message identifier, (4) the body of the message, (5) the sentiment label (bearish, bullish, or none) entered by the user, (6) a unique identifier of the user who posted the message, (7) the number of messages published by the user who posted the message, and (8) the number of followers of the user who posted the message. Our sentiment analysis builds on the first five features. The last three provide additional information on the network structure, which we briefly discuss in Appendix A.3.3.

Figure 3.2 shows a screenshot of the StockTwits website as of 3rd March 2020, for a query of AAPL, which is the ticker for Apple. The first message is labeled as bullish by the user "satkaru", the two next are unlabeled messages that will be classified by our machine learning algorithm, and the last message is labeled as bearish by the user "Etrading".

The left plot of Figure 3.3 shows the top 30 most discussed tickers on StockTwits. SPY, a large ETF that tracks the S&P 500 stock market index, is the most discussed ticker, followed by Apple and other big tickers. The messages about the 15 (30) biggest tickers represent 20% (25%) of the total number of messages, which indicates that users talk about a wide panel of tickers and not only big firms. The right graph shows a histogram of the number of messages per ticker. The x-axis is log-scaled because due to extreme values the distribution is highly skewed.

Text messages need to be transformed into a quantitative vector to be fed into our machine learning classifier, which in turn computes a sentiment score. This transformation consists of several steps. First, we apply some preprocessing operations to the text messages: an apostrophe handler, a contraction form handler (e.g., "aren't" becomes "are not"), tickers

[3] A message may refer to multiple tickers. We count any such message towards any ticker that it refers to. We give more information about this double counting in Appendix A.3.2.

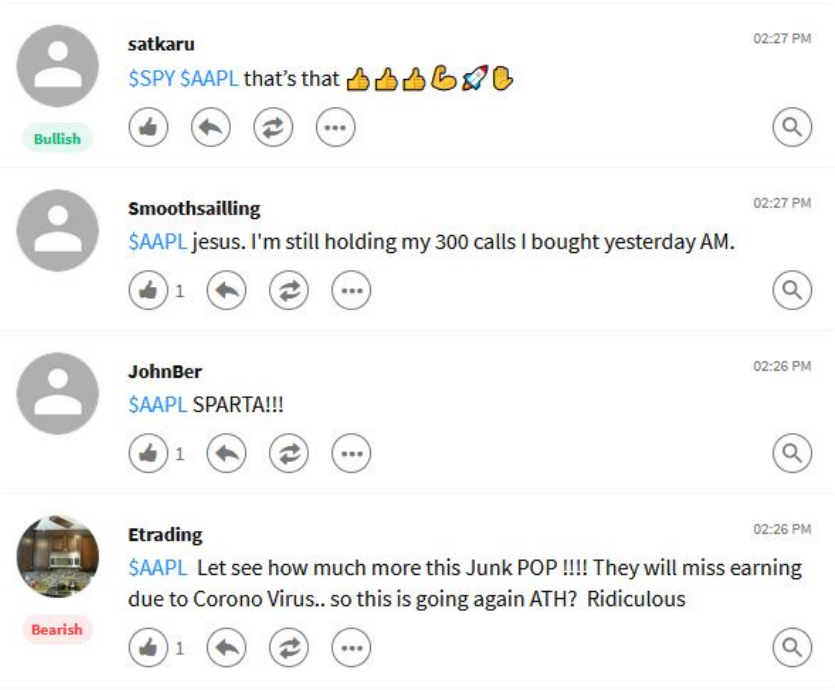


Figure 3.2: Screenshot of messages posted on StockTwits
Screenshot is from the 3rd of March 2020 for a query on AAPL, the ticker of Apple.

removal, stop words removal (e.g., “a", “the", “of")[4], users removal, lemmatization, URLs removal and a simple spell corrector dealing with more than two repeated characters (e.g., “soooo goooood" becomes “soo good"). Table 3.1 shows five examples of messages before and after preprocessing.

The next step is tokenization: the slicing of a text message into smaller units called terms or tokens. In financial lingo, some words only have meaning when associated with other words (i.e., “bad apple" or “bear flag"). N-gram models allow accounting for words frequently occurring together with other words. The main hyperparameter in an N-gram model is the number N of words that form a term: a unigram is a term with only one word, a bigram is a term with two consecutive words, etc. Larger N-gram models increase dramatically the size of the vocabulary (i.e., the collection of all terms considered). We select $N = 1,2,3$ and truncate the resulting vocabulary such that it consists of the one million most frequent terms in the union of all unigrams, bigrams and trigrams.

Figure 3.4 represents the bullish and bearish word clouds. These represent the most frequent terms in all user-labeled bullish and bearish messages relative to their total appearance, respectively. The size of the terms represents their relative weight in the cloud. In the bullish cloud, we see terms such as “bullish divergence", “room to grow", “lot potential" which we can clearly interpret as bullish signals. In the bearish cloud, we find terms such as “recent

[4]We follow Renault (2020) and Saif et al. (2014) and use a restrictive list of stopwords to avoid accuracy decrease.

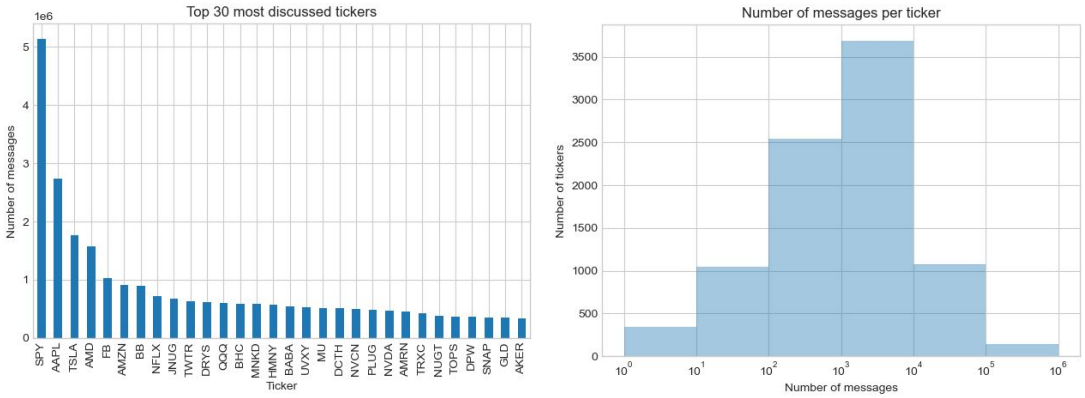


Figure 3.3: Ticker summary statistics
Left graph shows the top 30 most discussed tickers on StockTwits. SPY is the ticker of a large ETF tracking the S&P 500 and AAPL is the ticker for Apple. Right graph shows the distribution of the number of messages across tickers.

Figure 3.4: Word clouds
Bullish word cloud (left), bearish word cloud (right). These correspond to the most frequent terms (up to trigrams) in user-labeled bullish (bearish) messages relative to their total appearance. The size of the terms represents their importance in the cloud.

resistance", “short setup", “bad apple" which again we can directly interpret as bearish signals. These findings are reassuring in the sense that the content of the messages on StockTwits are consistent with their labels. We checked for anomalies at random, but did not find significant issues. Appendix A.3.4 discusses two such anomalies.

3.3 Sentiment Classification

The left plot of Figure 3.5 shows the proportions of user sentiment-labeled messages across time. In the early years of the platform, most messages were unlabeled, presumably because users were not familiar with the sentiment label yet. Albeit the proportion of unlabeled messages monotonically declines over the years, almost 60% of the more recent messages are still unlabeled. Overall, around 30 million messages are user-labeled and 60 million messages are unlabeled. We conjecture that by far not all unlabeled messages reflect market neutral opinions. Indeed, the right plot of Figure 3.5 reveals that a substantial part of user-unlabeled messages is machine learning classified as bullish or bearish. Hence these user-unlabeled

Before preprocessing	
(1)	@CassandraTwit $uvxy contango 3.5%...still long. gooooooood
(2)	$FRPT Take profits while you still can.
(3)	$UVXY $tvix go time boys and girls. Holding overnight again
(4)	$dnr Nice upgrade as company goes into its quiet period!
(5)	$SPY market won't reverse again towards closing. Get put options.
After preprocessing	
(1)	contango still long good
(2)	take profit while you still can
(3)	go time boy and girl hold overnight again
(4)	nice upgrade as company go into its quiet period
(5)	market will not reverse again towards closing get put options

Table 3.1: Preprocessing of five sample messages
Preprocessing operations include: punctuation removal, lower casing, apostrophe handling, contraction form handling (i.e., "won't" becomes "will not"), tickers removal, users removal, URLs removal, parsing and a simple spell corrector dealing with more than two repeated characters (i.e., "goooood" becomes "good")

messages contain indeed market relevant information, which we are able to capture by our algorithm.

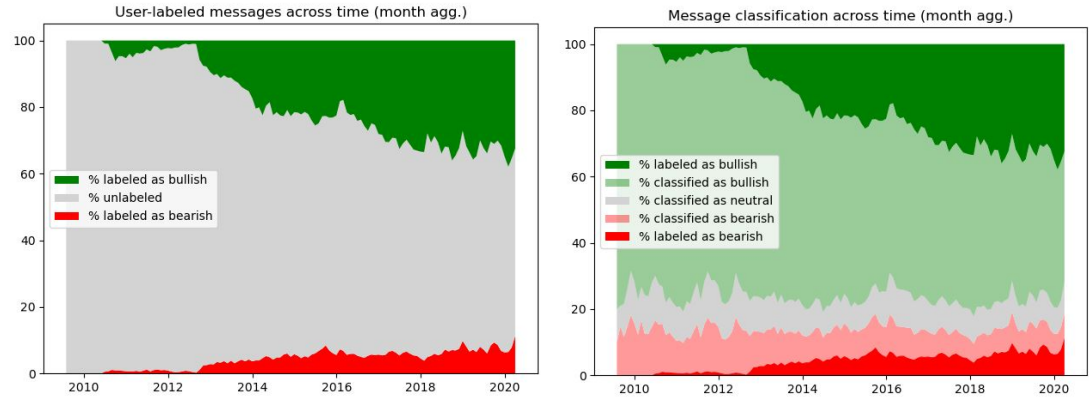


Figure 3.5: Message classification
Left plot shows the proportions of user-labeled messages: bullish (green), bearish (red), and unlabeled (gray) across time. The right plot shows the proportions of machine learning classified messages: bullish (light green predicted, green user-labeled), bearish (light red predicted, red user-labeled), and neutral (gray) across time. Proportions are aggregated monthly.

Among the user-labeled messages we find five times more bullish than bearish ones. This ratio indicates that investors are on average optimistic about the market, which is consistent with findings in the literature, e.g., Renault (2017). Such an imbalance is a well-known issue in machine learning classification as it creates a bias towards the over-represented class (see Chawla et al. (2004)). There are various ways to tackle class imbalance. An all-purpose standard approach in machine learning is to over-sample the minority class, which consists of randomly re-sampling from the minority class and thus artificially re-balance the class sizes

in the data. We use a different approach, which is tailored for our setup. As not every message carries a substantial stock market relevant sentiment, we deviate from the bullish–bearish dichotomy. Instead, we create an auxiliary neutral sentiment class to account for messages that do not take a clear stand. See Appendix A.3.7 for examples of such neutral messages.

We then run two independent binary classifiers. The first (second) classifies messages as bullish versus non-bullish (bearish versus non-bearish). We aggregate the two binary outcomes and classify a message as bullish (bearish) for the concordant combination bullish/non-bearish (non-bullish/bearish), and neutral otherwise. This approach is very simple and efficient, and eliminates the class imbalance bias at the same time. It builds on any traditional binary classifiers. We use logistic regression on Term Frequency-Inverse Document Frequency (TFIDF)-vectorized messages, as in, e.g., Yildirim et al. (2018), Qasem et al. (2015), Erdemlioglu et al. (2017). TFIDF is a widely used method to transform a text, in our case a message m, into a numerical vector, $TFIDF_m$. The dimension of this vector is equal to the size of the vocabulary (the collection of all terms across all messages). The components of the vector encode the importance of the corresponding terms t in the message m, as formally defined by $TFIDF_{m,t} = TF_{m,t} \cdot IDF_t$. The first factor measures how frequently term t appears in the message,

$$TF_{m,t} = \frac{\sum_{i=1}^{N_m} \mathbf{1}_{t=t_{m,i}}}{N_m}, \tag{3.1}$$

where N_m denotes the number of terms $t_{m,i}$ in message m. The second factor measures how important term t is to the message,

$$IDF_t = \log\Big(\frac{V}{\sum_{j=1}^{V} \mathbf{1}_{t\in m_j}}\Big), \tag{3.2}$$

where V denotes the total number of messages m_j. A term t appearing in many documents (such as “the", “is", “of") is likely to have low information content, hence a low IDF_t.

We use 80% of the user sentiment-labeled messages as a training set and keep 20% as a test set, then we run two binary classifiers. The first (second) classifier sets bullish (bearish) as positive and non-bullish (non-bearish) as negative class. Every message then classifies into one of the following combinations: (non-bullish, bearish), (bullish, bearish), (non-bullish, non-bearish), (bullish, non-bearish). For the first and last combinations, the two algorithms agree and the final classification is defined to be bearish (non-bullish, bearish) or bullish (bullish, non-bearish), respectively. For the two middle combinations, (bullish, bearish) and (non-bullish, non-bearish), the two algorithms disagree, so that the final classification is defined to be neutral. Formally, every message m is mapped onto either

$$m \mapsto \begin{cases} \text{(non-bullish, bearish)} & =: \text{bearish} \\ \text{(bullish, bearish)} & =: \text{neutral} \\ \text{(non-bullish, non-bearish)} & =: \text{neutral} \\ \text{(bullish, non-bearish)} & =: \text{bullish.} \end{cases} \tag{3.3}$$

To select optimal classification thresholds, we maximize the F1 scores. The F1 scores of the two binary classifiers differ because they depend on which class is defined as the positive one. We recap the definition of the F1 score in Appendix A.3.6. Figure 3.6 shows the F1 scores as functions of the threshold. Circles indicate the maximal F1 scores, along with the corresponding optimal thresholds, 0.50 and 0.72, respectively.

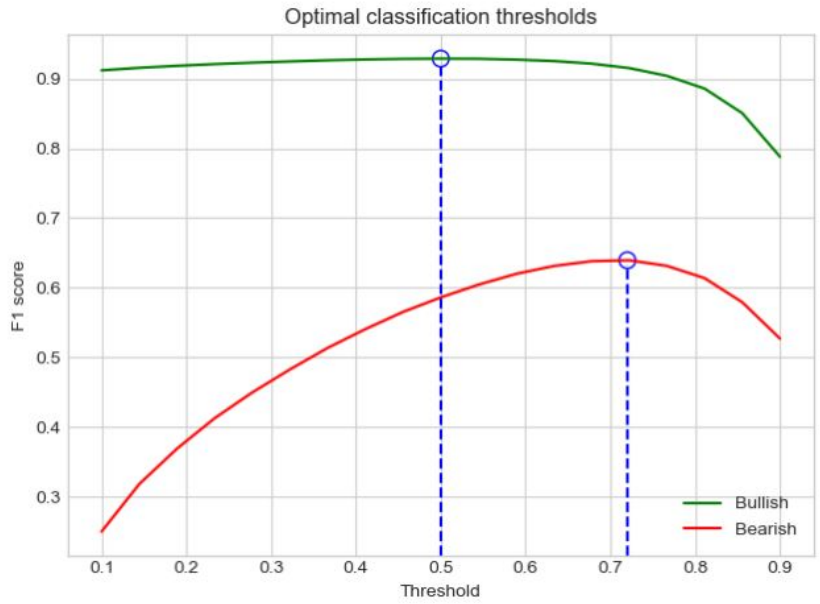


Figure 3.6: Optimal classification thresholds
The green (red) line is the F1 score for the bullish versus non-bullish (bearish versus non-bearish) classifier as a function of the threshold. Circles indicate the maximal F1 scores.

If the sentiment score of a message is bigger (smaller) than 0.72 (0.50), respectively, then both classifiers agree and the sentiment of the message is classified as bullish (bearish), respectively. If the sentiment score is between 0.50 and 0.72, the classifiers disagree, (bullish, bearish), and we consider the message as neutral. Finally, we overwrite the predicted sentiment of any message by the user-labeled sentiment whenever the latter is available. Research in sentiment classification shows that human annotators tend to agree about 80 to 85% of the time when evaluating the sentiment of a document (see, e.g., Wilson et al. (2005) and Chen et al. (2020)). This is a benchmark for the accuracy that a sentiment classifier should meet or beat. The out-of-sample accuracy of our combined classifier is 85.9%. Appendix A.3.6 provides in-sample and out-of-sample confusion matrices for our combined classifier.

The right plot of Figure 3.5 shows the proportions of our machine learning classifications across time. Percentages of bearish (user-labeled and classified as bearish) and bullish (user-labeled and classified as bearish) messages are stable over time, suggesting that our classification method is robust. Even if most messages were not user-labeled in the early years of the platform, as seen in the left plot of Figure 3.5, we are now able to classify the sentiment of all messages in the sample, including a neutral class. Consistent with the over-representation of

bullish messages observed in the user-labeled messages, there are substantially more messages classified as bullish than bearish. Examples of classified messages are given in Appendix A.3.7.

3.4 Polarity

We next aggregate message sentiments on a daily ticker-level and across the market. Thereto, we denote $C_{i,t,j} = 1, 0, -1$ for bullish, neutral, bearish, respectively, the sentiment of the jth message about ticker i on day t.[5] We follow Ranco et al. (2015) and define the sentiment polarity of ticker i on day t as

$$P_{i,t} = \frac{\sum_{j=1}^{V_{i,t}} (\mathbf{1}_{C_{i,t,j}=1} - \mathbf{1}_{C_{i,t,j}=-1})}{\sum_{j=1}^{V_{i,t}} (\mathbf{1}_{C_{i,t,j}=1} + \mathbf{1}_{C_{i,t,j}=-1})}, \tag{3.4}$$

where $V_{i,t}$ denotes the number of messages about ticker i on day t.[6]

As an aggregate, we define the market polarity as a weighted average over all tickers

$$P_t^M = \frac{\sum_i V_{i,t} \cdot P_{i,t}}{V_t^M}, \tag{3.5}$$

where $V_t^M = \sum_i V_{i,t}$ denotes the number of messages on day t. Figure 3.7 shows a scatter plot of the market polarity P_t^M versus the polarity of SPY. The slope coefficient of the regression line is statistically significantly positive and the contemporaneous Pearson correlation coefficient is 0.53, suggesting that the market polarity is an accurate measure of the aggregated sentiment of the market.[7] Also, consistent with Figure 3.5, SPY and market polarities are bullish-biased.

For the following time series analysis and event study we restrict our sample. There are two reasons for doing so. First, we keep computational cost at a reasonable level. Second, and more importantly, the time series of ticker-individual polarities exhibit spikes and are too noisy if the daily message volumes $V_{i,t}$ are too small. We therefore exclude from now on tickers whose median of daily message volume is less than 50. Our reduced sample contains 19 tickers, covering a range of security types, sectors, and market capitalization. We refer to Appendix A.3.5 for more details on the coverage, including summary statistics and the list of tickers covered.

To understand how our polarity measure is related to investor sentiment, we run linear

[5] We follow the close-to-close convention. First, we remove all non-business days from the sample, whereby messages posted on non-business days count towards the next business day. "Day t" then stands for the time interval from 4pm on the previous business day $t-1$ to 4pm on business day t. This convention is consistent with the stock return data, which are close-to-close, and thus avoids any look-ahead bias of our sentiment polarity.

[6] If $V_{i,t} = 0$ then we set $P_{i,t} = 0$.

[7] We do not expect P_t^M to be equal to the SPY polarity because the underlying sets of stocks differ: market polarity contains stocks that are not in the S&P500 and vice versa.

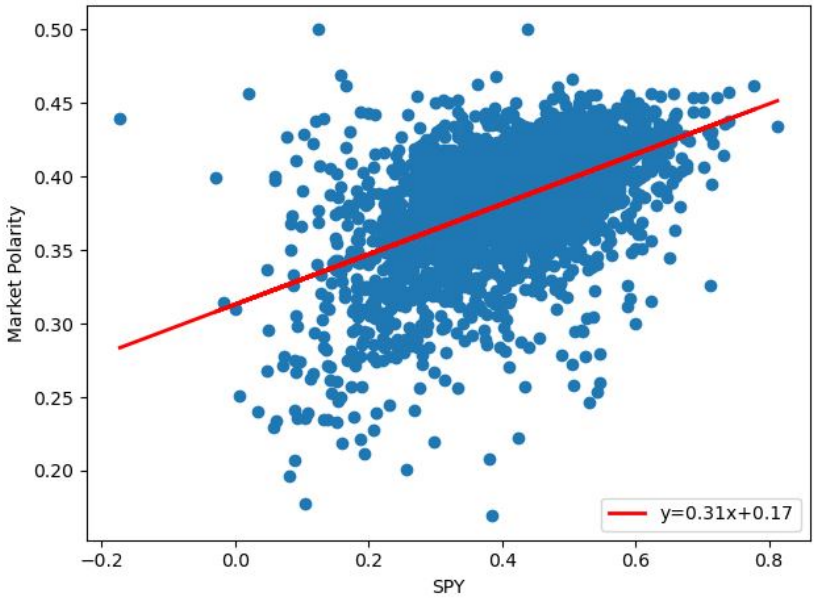


Figure 3.7: Market polarity versus SPY polarity
The red line shows the linear regression line and coefficients.

regressions of contemporaneous and next-day returns on polarity:

$$R_{i,t} = \alpha^{cont} + \beta^{cont} \cdot P_{i,t} + \epsilon_{i,t}^{cont}, \tag{3.6}$$

$$R_{i,t+1} = \alpha^{next} + \beta^{next} \cdot P_{i,t} + \epsilon_{i,t}^{next}. \tag{3.7}$$

Table 3.2 shows that the regression coefficient is positive and significant for contemporaneous returns. This indicates that polarity is a good contemporaneous proxy for the sentiment of investors. Further supporting evidence is given by the correlation between polarity and contemporaneous returns at the ticker level.

	$R_{i,t}$	$R_{i,t+1}$
Constant	-0.0047***	-0.0002
	(0.000)	(0.000)
$P_{i,t}$	0.009***	0.0003
	(0.000)	(0.000)
R^2	0.012	0.000
No. Obs.	34100	34100

Table 3.2: Regressions of returns on polarity
Results from linear regressions of contemporaneous and next-day stock returns on polarity. Stock returns are trimmed at the 5% percentile on both sides. Standard errors are reported in parentheses. Statistical significance at the 99%, 95%, and 90% level is indicated with ***, **, *, respectively.

Figure 3.8 shows the time series during 2019 for the top 6 most discussed tickers. In our entire sample of tickers, correlations are always positive and range between 0.1 and 0.3. In contrast, regressing next-day returns reveals that polarity has no predictive power for next-day stock returns unconditionally. In the following section we show how polarity has predictive power around specific events.

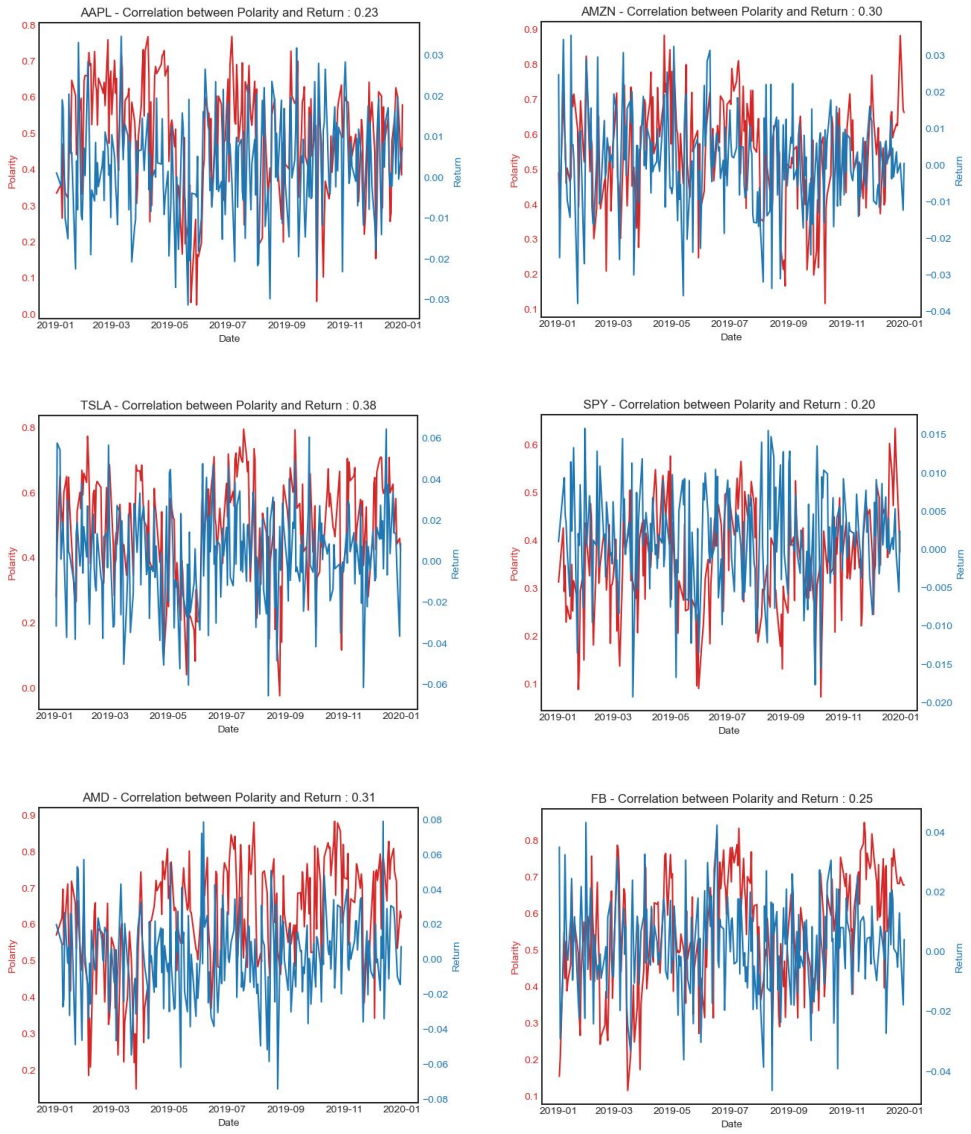


Figure 3.8: Correlation between return and polarity
Time series of daily polarity (red - left axis) and daily stock returns (blue - right axis) since 1st of January 2019 for the top 6 most discussed tickers. Pearson correlation between the two time series is shown in the title.

3.5 Event Study

Event studies constitute a statistical method widely used in financial econometrics (see, e.g., MacKinlay (1997)). In general, they are used to measure the effect of events on the market value of stocks. Well-known applications of event studies include the testing of various forms of the efficient market hypothesis (EMH) (see Fama et al. (1969) and Fama (1991)).

3.5.1 Events

We define events as days with an unusual large number of messages for individual tickers. We conjecture that a sudden peak in StockTwits message volume indicates that an important corporate or stock market event is happening on the day of the peak. Figure 3.9 shows that increases (decreases) in message volumes are positively associated with increases (decreases) in contemporaneous weekly stock transaction volumes. These co-movements indicate that investors who post messages about stocks also trade them accordingly, adjusting their portfolios. This suggests that message volume peaks are a good proxy for corporate and stock market events.

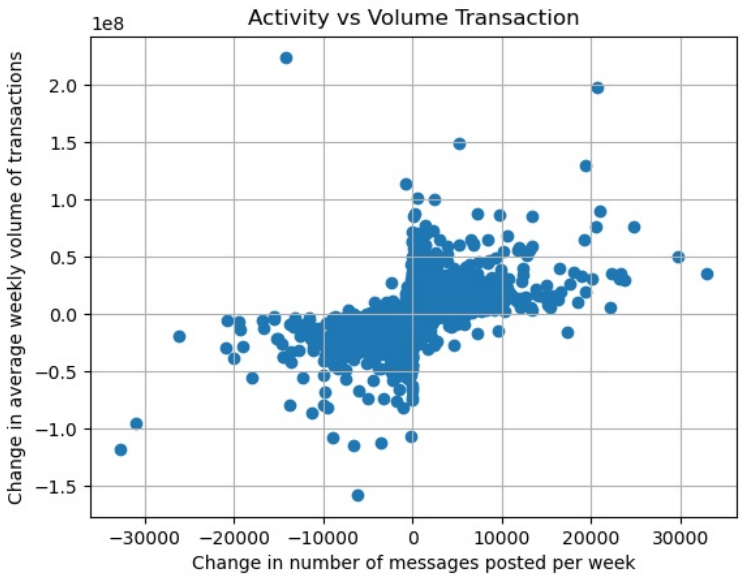


Figure 3.9: Message activity and transaction volume
Changes in weekly volume of transactions on the y-axis versus changes in message activity on the x-axis. Activity is measured in weekly messages posted per ticker.

To measure unusual activity peaks, we use as benchmark model a one-year rolling window regression of daily relative message volume changes of ticker i on daily relative total message volume changes.[8] Formally,

$$\frac{\Delta V_{i,t}}{V_{i,t-1}} = \alpha_i^V + \beta_i^V \cdot \frac{\Delta V_t^M}{V_{t-1}^M} + \epsilon_{i,t}, \tag{3.8}$$

which gives the one-year rolling estimates $\hat{\alpha}_i^V$ and $\hat{\beta}_i^V$. We then define the abnormal message volume changes for ticker i on day t as

$$AV_{i,t} = \frac{\Delta V_{i,t}}{V_{i,t-1}} - \left(\hat{\alpha}_i^V + \hat{\beta}_i^V \cdot \frac{\Delta V_t^M}{V_{t-1}^M} \right). \tag{3.9}$$

We define an event for ticker i as any day t where the standardized abnormal volume exceeds two,

$$\frac{AV_{i,t} - \hat{\mu}_{AV_i}}{\hat{\sigma}_{AV_i}} > 2, \tag{3.10}$$

where $\hat{\mu}_{AV_i}$ and $\hat{\sigma}_{AV_i}$ denote the one-year rolling empirical mean and standard deviation.

Next we define the type of the event as either bullish, neutral or bearish. We use the abnormal polarity $AP_{i,t}$ of the event date to assess how on average investors perceive the event. Figure 3.10 shows the distribution of abnormal polarities on event dates. We chose to use the one-

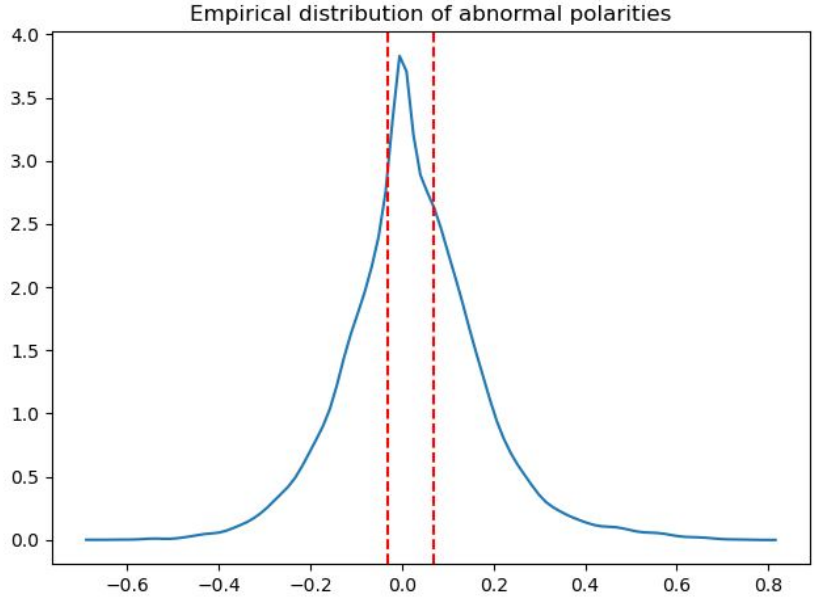


Figure 3.10: Empirical distribution of abnormal polarities on event dates
The red dashed lines show the one-third and two-third percentiles.

[8] Accounting for the lead time of the one-year rolling estimation window, the event study effectively applies to the shorter period from January 2011 to March 2020.

third (-0.03) and two-third percentile (0.07) of the distribution of abnormal polarities as thresholds for the type of the event. We define the type of the event for ticker i at t as

$$Type_{i,t} = \begin{cases} Bullish & \text{if } AP_{i,t} > 0.07, \\ Neutral & \text{if } AP_{i,t} \in [-0.03, 0.07], \\ Bearish & \text{if } AP_{i,t} < -0.03. \end{cases} \tag{3.11}$$

Overall, across 19 tickers, we identify 1131 events, whereof 454 bullish, 294 neutral, and 383 bearish types. This coverage is on par with previous studies (e.g., MacKinlay (1997) analyze 30 stocks and 600 events). Figure 3.11 shows the aggregate events and their types across time. The count of events looks stationary over time, apart from a build up phase of the platform in the early part until 2014. The distribution of event types is also balanced across time.

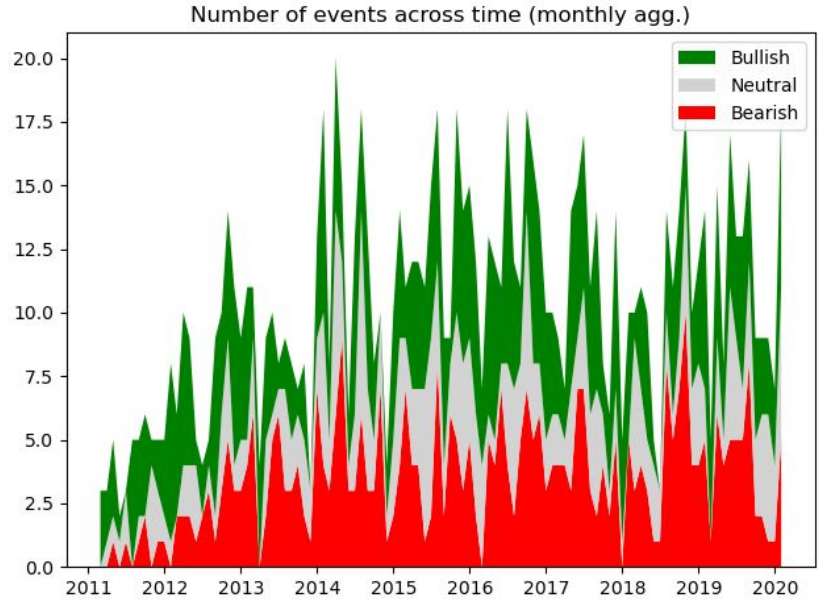


Figure 3.11: Number of events of each type across time
The green area shows bullish events, the gray area shows neutral events and the red area shows bearish events. Numbers are aggregated monthly.

As an illustration, Figure 3.12 shows for Apple the time series of message volume and the corresponding events. Between January 2011 and March 2020, our algorithm identified 73 events for Apple. What are these events? Remarkably, we capture a variety of corporate events and disclosures. Earning announcements constitute about half of the events. Other events include Apple Keynotes (presentations that Apple gives to the press, often presenting new products), or CEO letters addressed to investors. Table 3.3 lists a few examples.

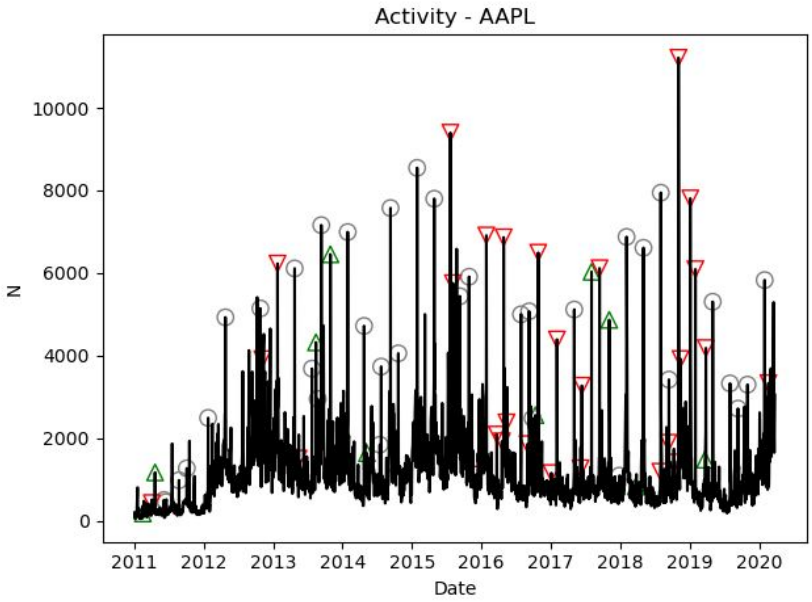


Figure 3.12: Daily message volume for Apple
Events are days with an unusual high number of messages. The green upper-triangles show bullish events, gray circles are neutral events and red down-triangles represent bearish events.

3.5.2 Abnormal Stock Returns

How do stock returns behave around events? Similar to the relative message volume changes, we use as benchmark model a one-year rolling window regression of the daily returns of ticker i on the daily market returns, R_t^M, i.e., daily excess returns of the S&P500,

$$R_{i,t} = \alpha_i^R + \beta_i^R \cdot R_t^M + \epsilon_{i,t}, \tag{3.12}$$

which gives the one-year rolling estimates $\hat{\alpha}_i^R$ and $\hat{\beta}_i^R$. This implies the abnormal returns

$$AR_{i,t} = R_{i,t} - \left(\hat{\alpha_i}^R + \hat{\beta_i}^R \cdot R_t^M\right). \tag{3.13}$$

We define the cumulative abnormal returns (CAR) around a ticker i event τ as

$$CAR_i(\tau, t) = \sum_{s=-20}^{t} AR_{i,\tau+s}, \tag{3.14}$$

and the cumulative average abnormal returns (CAAR) across all $N = 1131$ events as

$$CAAR(t) = \frac{1}{N} \sum_{j=1}^{N} CAR_{i_j}(\tau_j, t). \tag{3.15}$$

Left plot of Figure 3.13 shows the CAAR around the events. This plot is consistent with MacKinlay (1997). It shows that CAAR related to bearish (bullish) events exhibits a significant

Date	Description	Type
2012-04-24	Earnings announcement	Bullish
2012-09-12	Presents iPhone 5	Neutral
2014-09-09	Presents Apple Watch	Neutral
2017-05-02	Announces drop in iPhone sales	Bearish
2017-08-31	Earnings announcement	Bullish
2017-09-12	Presents iPhone X	Bearish
2019-01-02	CEO Letter to investors	Bearish
2019-09-10	Presents iPhone 11	Neutral
2019-10-30	Earnings announcement	Bullish

Table 3.3: Selected events and associated description and types for Apple
This list is for illustration and non-exhaustive (9 out of 73).

downward (upward) jump at the event date, respectively. These jumps are followed by a flat CAAR during the 20 days after the event. Interestingly, there is a systematic shift in the CAAR already 1 day before the event. However, this shift is relatively small compared to the jump on the event day: one day before the event, the bullish (bearish) CAAR equals 0.019 (-0.023). The CAAR related to the neutral events exhibits a slight upward shift around the event date but it fades away after a few days. The CAAR related to bearish events shifts already a few days before the event but this shift is not statistically significant. This is in line with Figure 3.14, which shows that the CAR distributions prior to the events are not significantly different from zero. This is confirmed by the Mann-Whitney U-tests shown in Table 3.4. CAR has no predictive power on the type of the event: five days before an event, the median of the CAR distribution of the bullish events is not statistically different from the median of the neutral events. The same holds for the bearish events.

3.5.3 Abnormal Polarity

How does sentiment polarity behave around events? Similar to the above, we use as benchmark model a one-year rolling window regression of the daily polarity of ticker i on the daily market polarity defined in (3.5),

$$P_{i,t} = \alpha_i^P + \beta_i^P \cdot P_t^M + \epsilon_{i,t}, \tag{3.16}$$

which gives the one-year rolling estimates $\hat{\alpha}_i^P$ and $\hat{\beta}_i^P$. This implies the abnormal polarity

$$AP_{i,t} = P_{i,t} - \left(\hat{\alpha_i}^P + \hat{\beta_i}^P \cdot P_t^M\right). \tag{3.17}$$

We define the cumulative abnormal polarity (CAP) around a ticker i event τ as

$$CAP_i(\tau, t) = \sum_{s=-20}^{t} AP_{i,\tau+s}, \tag{3.18}$$

and the cumulative average abnormal polarities (CAAP) across all $N = 1131$ events as

$$CAAP(t) = \frac{1}{N} \sum_{j=1}^{N} CAP_{i_j}(\tau_j, t). \tag{3.19}$$

The right plot of Figure 3.13 shows the CAAP around the events. There are two main findings. First, in contrast to CAAR, the CAAP for bullish and bearish events is not constant after the event date, suggesting that users' sentiments about stocks tend to be biased towards recent past events. A possible explanation is that users might still post bullish (bearish) messages about a bullish (bearish) event during several days after the event. This is in contrast to the returns that immediately normalize after the event. Second, and more interestingly, the CAAP for bullish and bearish events shifts several days earlier than the CAAR. This indicates that investors are on average able to anticipate the type of an event in the near future. However, this sentiment only manifests through the social media, but not through abnormal returns.

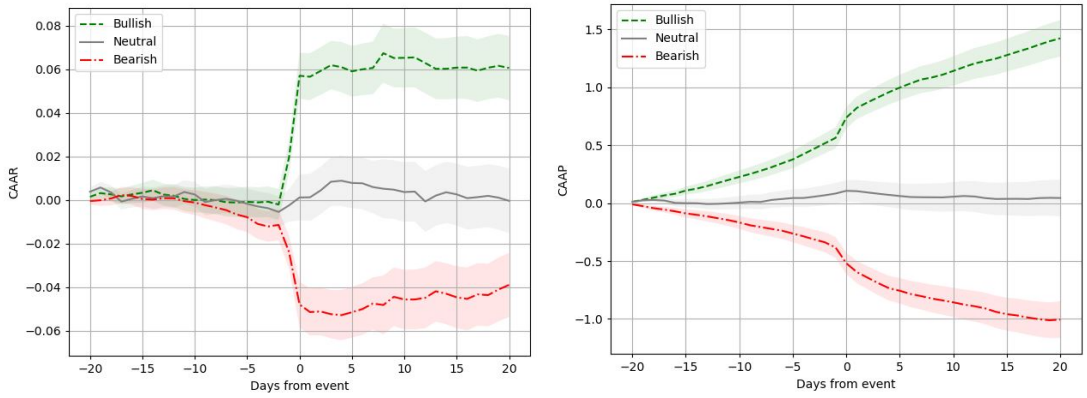
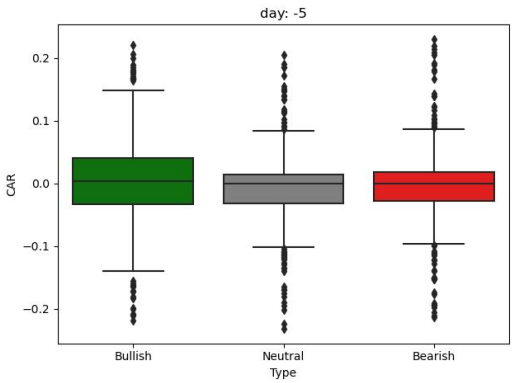
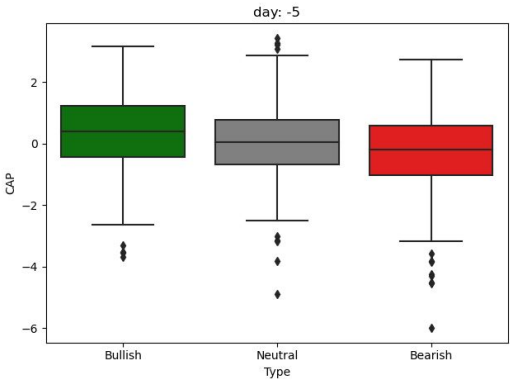


Figure 3.13: CAAR and CAAP around identified events
Cumulative average abnormal returns (left plot) and cumulative average abnormal polarity (right plot) around identified events. CAAR and CAAP related to bearish, neutral and bullish events are displayed with the red, gray and green line, respectively. Areas around lines show confidence intervals at the 95% level.

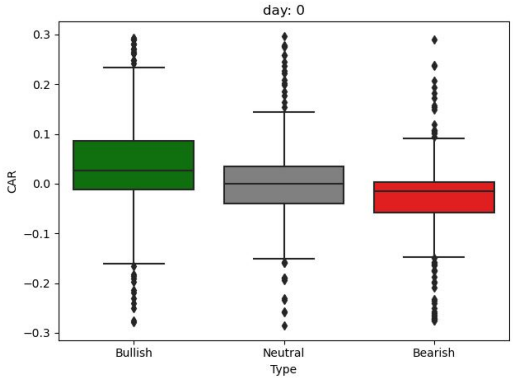
Figure 3.14 illustrates this striking finding with box plots (see Dekking et al. (2005) and Tukey (1977)) showing the distributions of the CAR and CAP, for all three event types, 5 days before the event, at the event date, and 5 days after the event, respectively.



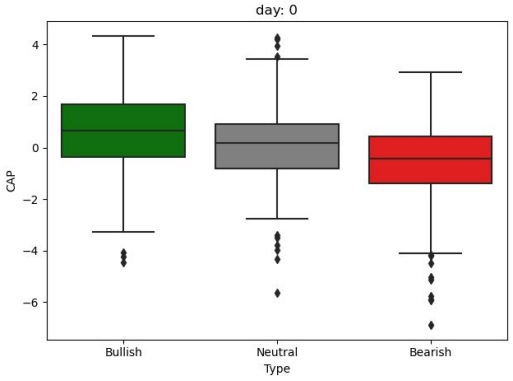
CAR 5 days before the event



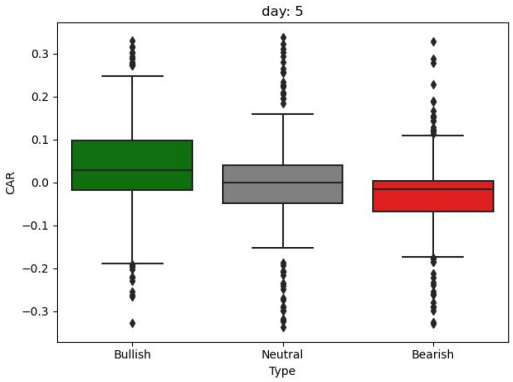
CAP 5 days before the event



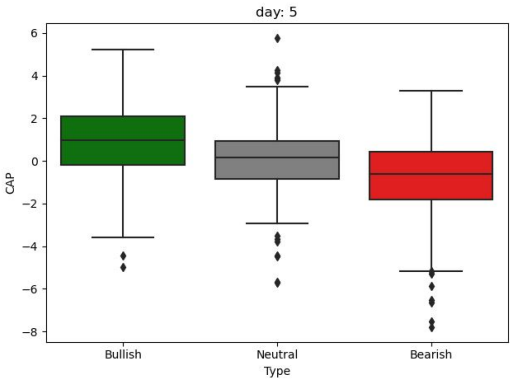
CAR on the event date



CAP on the event date



CAR 5 days after the event



CAP 5 days after the event

Figure 3.14: Distributions of CAP and CAR around events
The line inside a box shows the median while the edges of each box represent the 25% and 75% quantile of the distribution. From above the edges of a box, a distance of 1.5 times the interquartile range is measured and a whisker is drawn up to the largest and lowest observed point from the data that falls within this distance. Interquartile range is equal to the third quartile minus the first quartile.

CAR					
	Alternative Hypothesis	U	Z	n_1	n_2
τ-5	$H_1 : \theta_{bullish} > \theta_{neutral}$	61109	-1.70	452	292
	$H_1 : \theta_{neutral} > \theta_{bearish}$	54168	-0.52	292	380
τ	$H_1 : \theta_{bullish} > \theta_{neutral}$	50967	-5.25***	452	292
	$H_1 : \theta_{neutral} > \theta_{bearish}$	43100	-4.96***	292	380
τ+5	$H_1 : \theta_{bullish} > \theta_{neutral}$	52738	-4.63***	452	292
	$H_1 : \theta_{neutral} > \theta_{bearish}$	43239	-4.91***	292	380
CAP					
	Alternative Hypothesis	U	Z	n_1	n_2
τ-5	$H_1 : \theta_{bullish} > \theta_{neutral}$	55408	-3.70***	452	292
	$H_1 : \theta_{neutral} > \theta_{bearish}$	47998	-3.00***	292	380
τ	$H_1 : \theta_{bullish} > \theta_{neutral}$	49385	-8.98***	452	292
	$H_1 : \theta_{neutral} > \theta_{bearish}$	42101	-5.36***	292	380
τ+5	$H_1 : \theta_{bullish} > \theta_{neutral}$	44364	-7.55***	452	292
	$H_1 : \theta_{neutral} > \theta_{bearish}$	40515	-6.00***	292	380

Table 3.4: Mann-Whitney U-test statistics
Mann-Whitney U-test statistics for pairwise significant differences between distribution medians. Under the null hypothesis, the two samples represent two distributions with equal median values. Statistical significance at the 99%, 95%, and 90% level is indicated with ***, **, *, respectively.

To check statistical significance, we use the Mann-Whitney U-test (see Mann and Whitney (1947) and Sheskin (1998)) to test whether the three samples (bullish, neutral and bearish) represent populations with different median values.[9] Table 3.4 shows U-test statistics for pairwise comparisons. The null is rejected in every case except for CAR at $\tau - 5$. That is, 5 days before the event, CAR has no predictive power on the type of event. This is consistent with the EMH. In contrast, 5 days before the event, CAP can predict the type of event. At the event date, the medians of the CAR shift as the abnormal returns jump for both bullish and bearish events. This is also consistent with the EMH. Finally, 5 days after the event, the distributions of the CAR are very similar to the ones at the event date. Again, this is consistent with the EMH, as all new information is instantaneously embedded into the prices and the returns normalize after the event, immediately. The medians of the CAP 5 days after the event exhibit an extended shift compared to the ones at the event date, as investors continue to post about recent past events.

[9] This interpretation only holds under stringent assumptions on the populations, namely that the two population distributions are equal up to a shift. Under the null hypothesis, the three samples represent distributions with equal medians. Let θ_i be the median of the distribution i. Formally, we test $H_0 : \theta_{bullish} = \theta_{neutral}$ against $H_1 : \theta_{bullish} > \theta_{neutral}$ and $H_0 : \theta_{neutral} = \theta_{bearish}$ against $H_1 : \theta_{neutral} > \theta_{bearish}$ 5 days before an event, on event date and 5 days after an event. We define U as the Mann-Whitney test statistic, Z as the normal approximation of the Mann-Whitney test statistic for large sample sizes, n_1 and n_2 as the sample sizes. We refer to Sheskin (1998) for the test statistic computation.

3.6 Sentiment-Sorted Portfolios

We assess the economic relevance of the sentiment polarity and construct sorted portfolios. Thereto, we define for every ticker i and day t

$$CAP_{i,t} = \sum_{s=t-14}^{t} AP_{i,s}, \tag{3.20}$$

which is the running CAP over the last 14 days plus the current day t (we rebalance the portfolio at the close on day t).[10] Note the difference to (3.18). While we cannot predict the arrival of an event, we assume that the more $CAP_{i,t}$ deviates from zero the more likely there will be an event on the next day. We will thus use $CAP_{i,t}$ as a baseline signal for market timing. However, as we have seen above, CAP continues to shift after an event. To avoid exposures to short-term reversals, we thus reset the running CAP after every event. Formally, let $\tau_{i,t} \leq t$ denote the most recent past event date by t of ticker i. Then we define the reset CAP

$$CAP^{(R)}_{i,t} = \sum_{s=\max\{t-14,\tau_{i,t}+1\}}^{t} AP_{i,s} = \begin{cases} CAP_{i,t}, & \text{if } \tau_{i,t} < t-14, \\ \sum_{s=\tau_{i,t}+1}^{t} AP_{i,s}, & \text{if } t-14 \leq \tau_{i,t} < t, \\ 0, & \text{if } \tau_{i,t} = t, \end{cases} \tag{3.21}$$

where we used the convention that $\sum_{s=t+1}^{t} \cdot = 0$.

We also define time-varying thresholds on the reset CAP for market timing. For every day t, we estimate the mean μ_t and standard deviation σ_t of $CAP^{(R)}_{i,t}$ across the 19 tickers i. For a fixed multiplier x, we define $U_t(x) = \mu_t + x \cdot \sigma_t$ the upper threshold, and $L_t(x) = \mu_t - x \cdot \sigma_t$ the lower threshold. Figure 3.15 shows the time series of the cross-sectional mean μ_t and the 99% confidence interval, $L_t(x)$ and $U_t(x)$ for $x = 2.58$. As a robustness check of our approach, we observe that the mean is well centered at zero. We also see a regime change in early 2015. In the first regime the standard deviation is much larger (and more volatile) than in the second regime.[11] Appendix A.3.8 contains the results for the 95% ($x = 1.96$) and 99.5% ($x = 2.81$) confidence intervals.

Based on these signals, we now construct reset-CAP-sorted portfolios. Formally, we define the ticker sets $I^{bull}_t = \{i \mid CAP^{(R)}_{i,t} > U_t(x)\}$ and $I^{bear}_t = \{i \mid CAP^{(R)}_{i,t} < L_t(x)\}$. At the close of any day t, we form the equally weighted bullish (bearish) portfolio consisting of tickers in I^{bull}_t (in I^{bear}_t), and realize the 1-day returns. If any of the index sets is empty, we set the corresponding return to zero. The top-left plot of Figure 3.16 shows the cumulative log returns of bullish and bearish portfolios as well as the S&P500. Overall, the portfolio performance is consistent with our approach: the bullish (bearish) portfolio outperforms (under-performs) the market. Remarkably, the upward (downward) steps suggest that our portfolio strategy succeeds to take the right positions just before an event. The remaining plots of Figure 3.16

[10] As above, we work here with the restricted sample of 19 tickers and dates t ranging through all business days of the sample period, excluding the first 14 days (for the CAP) and the last day (for the last portfolio holding period).

[11] We could not find an exogenous cause for this regime change.

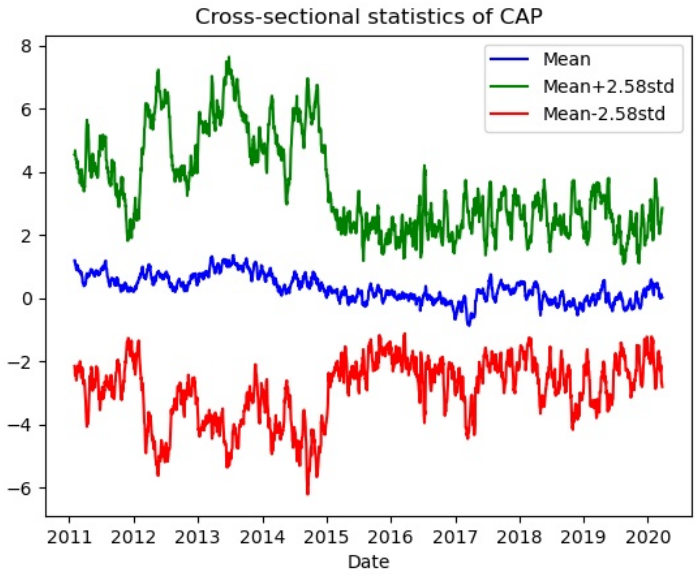
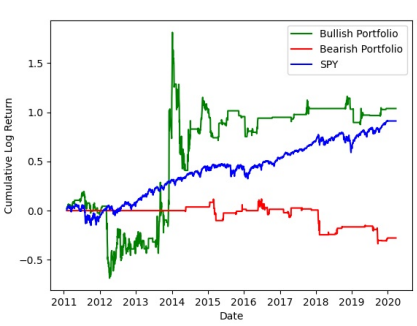
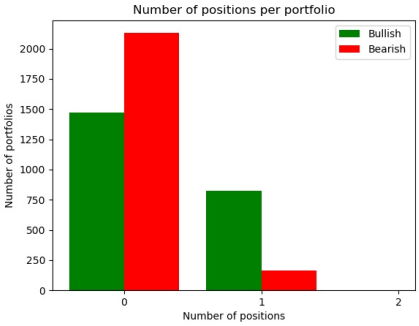


Figure 3.15: Cross-sectional statistics of $CAP_{i,t}^{(R)}$
The blue line shows the daily cross-sectional mean, the green (red) line shows the cross-sectional mean plus (minus) 2.58 standard deviations, respectively.

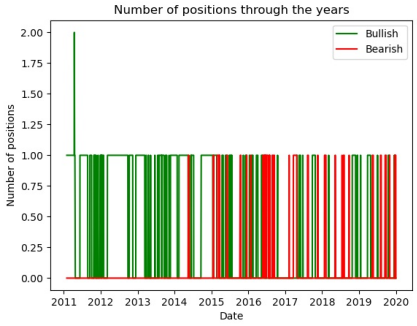
show the number of positions across time of our portfolios. Most of the returns are earned with portfolios consisting of very few tickers. This is a result of our market timing and stock picking strategy: we only invest in the top/bottom percentiles of CAP, whenever our signal is strong enough.



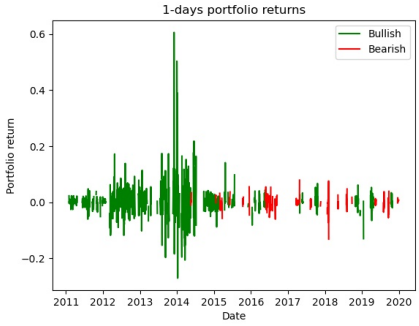
Cumulative log returns



Distribution of number of positions



Number of positions over time



Daily returns

Figure 3.16: Bullish and bearish portfolios for $x = 2.58$
Top left plot shows the cumulative log returns of the portfolios over the years, top right plot shows the distribution of the number of positions in the portfolios, bottom left plot is the number of positions over time and bottom right plot is the daily returns of both portfolios.

3.7 Conclusion

We extract a large sample of messages from StockTwits from January 2010 to March 2020, covering US and Canadian stocks. Messages are either user-labeled as bullish or bearish or left unlabeled. Using the user-labeled messages as training set, we run logistic regressions on TFIDF vectorized messages to classify all unlabeled messages as either bullish, neutral or bearish. We observe a 5-to-1 bullish-to-bearish ratio, indicating that investors are on average optimistic. We build time series of daily sentiment polarity for individual tickers and the aggregate market. We show that daily polarity is positively associated to contemporaneous stock returns, but this result loses its significance against next-day returns. We then define events as days with sudden peaks of message volume and relate them to corporate and stock market events. We show that cumulative abnormal polarity has significant predictive power on the type of event, in contrast to cumulative abnormal returns. We also note that investor sentiment about a ticker tends to be biased towards recent past events. As robustness check, we show that our event study on cumulative abnormal returns is consistent with previous literature on the efficient market hypothesis. The performance of sentiment-sorted portfolios illustrates the economic relevance of our sentiment measure.

Conclusion

The three chapters presented in this thesis use neural networks and natural language processing to develop or improve financial risk models.

The first chapter contributes to the literature on reduced-form models for multiperiod corporate default prediction using a doubly stochastic formulation. The probability of default in a specific horizon is a function of the forward default intensity and of the forward combined intensity. Previous literature proposes a model to predict corporate default at multiple horizons by estimating these forward intensities via maximum likelihood. To do so, they use a linear assumption in the relationship between the variables and the forward intensities. I show in the first chapter that a significant improvement is achieved by relaxing the linear assumption and using an artificial neural network instead. To allow comparison with the benchmark model, I choose to work with a similar set of features consisting of firm-specific accounting variables and macroeconomic variables. However, a growing area of study is building market sentiment measure and is showing their predictive power on the stock market. A potential and interesting venue for future research would be to gauge the predictive power of such features for default prediction.

The second chapter investigates the interconnections among a set of financial instutions. We propose a recurrent neural network approach to reduce the computational complexity of computing directed information. This approach is well-suited to infer the causal structure of large networks. Future research could focus on using this new methodology as a preliminary feature selection of another predictive model. Feature selection is a process used in machine learning which consists of keeping only a subset of relevant features to avoid overfitting or reduce dimensionality.

The third chapter classifies the sentiment of a large sample of social media messages to create stock-aggregate daily sentiment polarity measure. We show that conditionally on a stock market event happening, investors are on average able to anticipate the type of the event. Future research could focus on understanding the causes of this result at the user-level. In particular, some users have more influence than others because they either have more followers or they are more active on the platform. Are these users performing on average better than less active users ?

A Appendix

A.1 Appendix to Chapter 1

A.1.1 Relation between $g_{it}(\tau)$ and $\psi_{it}(\tau)$

The relation between $g_{it}(\tau)$ and $\psi_{it}(\tau)$ is given by :

$$\begin{aligned} g_{it}(\tau) &= \frac{F'_{it}(\tau)}{1-F_{it}(\tau)} \\ &= \psi_{it}(\tau) + \psi'_{it}(\tau)\tau. \end{aligned} \tag{A.1}$$

Proof. The last equation above can be computed with the first derivative of $\psi_{it}(\tau)$ with respect to time. Using Definition 1.8 we have :

$$\begin{aligned} \psi'_{it}(\tau) = \frac{u'v - uv'}{v^2} &= \frac{\frac{F'_{it}(\tau)}{1-F_{it}(\tau)}\tau + \ln(1-F_{it}(\tau))}{\tau^2} \\ &= \frac{\frac{F'_{it}(\tau)}{1-F_{it}(\tau)}}{\tau} + \frac{\ln(1-F_{it}(\tau))}{\tau^2} \\ \Rightarrow \psi'_{it}(\tau)\tau &= \frac{F'_{it}(\tau)}{1-F_{it}(\tau)} + \frac{\ln(1-F_{it}(\tau))}{\tau} \\ \Rightarrow \frac{F'_{it}(\tau)}{1-F_{it}(\tau)} &= \psi_{it}(\tau) + \psi'_{it}(\tau)\tau. \end{aligned}$$

□

A.1.2 Computation of $\psi_{it}(\tau)\tau$

The quantity $\psi_{it}(\tau)\tau$ that we are looking for is :

$$\psi_{it}(\tau)\tau = \int_0^\tau g_{it}(s)ds. \tag{A.2}$$

Proof. Integrating by parts $\psi'_{it}(s)s$ between 0 and τ, we have :

$$\begin{aligned}\int_0^\tau \psi'_{it}(s)\cdot s\cdot ds &= \psi_{it}(s)\cdot s|_{s=0}^{s=\tau} - \int_0^\tau \psi_{it}(s)\cdot 1\cdot ds \\ &= \psi_{it}(\tau)\tau - \int_0^\tau \psi_{it}(s)ds.\end{aligned}$$

Integrating both sides of equation A.1 leads to :

$$\begin{aligned}\int_0^\tau g_{it}(s)ds &= \int_0^\tau \psi_{it}(s)ds + \int_0^\tau \psi'_{it}(s)\cdot s\cdot ds \\ &= \int_0^\tau \psi_{it}(s)ds + \psi_{it}(\tau)\tau - \int_0^\tau \psi_{it}(s)ds \\ &= \psi_{it}(\tau)\tau.\end{aligned}$$

□

A.1.3 Likelihood function

The likelihood function has been developed in Duan et al. (2012). However, the likelihoods have to be slightly updated to be compatible with the neural network framework. One neural network is trained to compute f_{it} and the other is trained to output h_{it} where $g_{it} = f_{it} + h_{it}$. Let us denote λ and μ the set of parameters (weights) tuned in the neural network for f_{it} and h_{it} respectively. I impose non-negativity on both f_{it} and h_{it} to ensure that the combined exit intensity is at least bigger than the forward default intensity. The overall likelihood function for horizon of prediction τ is by definition given by

$$\mathscr{L}_\tau(\lambda,\mu;\tau_C,\tau_D,X) = \prod_{i=1}^{N}\prod_{t=0}^{T-1}\mathscr{L}_{\tau,i,t}(\lambda,\mu;\tau_{C_i},\tau_{D_i},X_{it}), \tag{A.3}$$

where

$$\begin{aligned}\mathscr{L}_{\tau,i,t}(\lambda,\mu;\tau_{C_i},\tau_{D_i},X_{it}) &= \mathbf{1}_{t_{0_i}\le t,\tau_{C_i}>t+\tau+1}\cdot\mathbb{P}_t(\tau_{C_i}>t+\tau+1) \\ &+ \mathbf{1}_{t_{0_i}\le t,\tau_{D_i}=\tau_{C_i}\le t+\tau}\cdot\mathbb{P}_t(t+\tau<\tau_{D_i}=\tau_{C_i}\le t+\tau+1) \\ &+ \mathbf{1}_{t_{0_i}\le t,\tau_{C_i}\le t+\tau,\tau_{D_i}\neq\tau_{C_i}}\cdot\mathbb{P}_t(t+\tau<\tau_{D_i}\neq\tau_{C_i}\le t+\tau+1) \\ &+ \mathbf{1}_{t_{0_i}>t} + \mathbf{1}_{\tau_{C_i}\le t},\end{aligned} \tag{A.4}$$

with

$$\mathbb{P}_t(\tau_{C_i} > t+\tau+1) = \exp(-\sum_{s=0}^{\tau} g_{it}(s)\Delta t), \tag{A.5}$$

$$\mathbb{P}_t(t+\tau < \tau_{D_i} = \tau_{C_i} \leq t+\tau+1) = \begin{cases} 1-\exp(-f_{it}(0)\Delta t), & \text{if } \tau_{C_i} = t+1 \\ \exp(-\sum_{s=0}^{\tau_{C_i}-t-2} g_{it}(s)\Delta t)\times & \\ \quad (1-\exp[-f_{it}(\tau_{C_i}-t-1)\Delta t], & \text{otherwise} \end{cases} \tag{A.6}$$

$$\mathbb{P}_t(t+\tau < \tau_{D_i} \neq \tau_{C_i} \leq t+\tau+1) = \begin{cases} 1-\exp(-g_{it}(0)\Delta t)- & \\ \quad (1-\exp(-f_{it}(0)\Delta t)), & \text{if } \tau_{C_i} = t+1 \\ \exp(-\sum_{s=0}^{\tau_{C_i}-t-2} g_{it}(s)\Delta t)\times & \\ \quad (\exp[-f_{it}(\tau_{C_i}-t-1)\Delta t- & \\ \quad \exp[-g_{it}(\tau_{C_i}-t-1)\Delta t], & \text{otherwise} \end{cases} \tag{A.7}$$

Since the indicator functions are all mutually exclusive, taking the log of the likelihood function is very helpful. After the log-linearization, the product terms become summation terms, and the two last indicator functions drop. We are left with the summation of the indicator functions times the logarithm of each probability defined above. Similar to the proposition 2 in Duffie et al. (2007) and subsection 3.2 in Duan et al. (2012), the pseudo log-likelihood is the product of seperate terms which are function of f and g. The first decomposition consists of separating terms involving f and terms involving g. The second decomposition consists of separating terms corresponding to different τ. In the end, we get two likelihood functions to estimate for each horizon. For each horizon the parameters of the likelihood function involving f are estimated in a neural network with output $N_{it}^{(\lambda)}$. The parameters of the likelihood function involving h are estimated in a neural network with output $N_{it}^{(\mu)}$. The combined exit forward intensity g is assumed to be of the form $f+h$ as in previous literature. Since $\exp(-f)-\exp(-g) = \exp(-f)-\exp(-f-h) = \exp(-f)*(1-\exp(-h))$, the first decomposition is the following :

$$\begin{aligned} \mathscr{L}_{\tau,i,t}(\lambda;\tau_{C_i},\tau_{D_i},X_{it}) = {} & \mathbf{1}_{t_{0_i}\leq t,\tau_{C_i}>t+\tau+1}\cdot\exp(-\sum_{s=0}^{\tau} f_{it}(s)\Delta t) \\ & + \mathbf{1}_{t_{0_i}\leq t,\tau_{D_i}=\tau_{C_i}\leq t+\tau}\cdot\exp(-\sum_{s=0}^{\tau_{C_i}-t-2} f_{it}(s)\Delta t)\cdot(1-\exp(-f_{it}(\tau_{C_i}-t-1)\Delta t) \\ & + \mathbf{1}_{t_{0_i}\leq t,\tau_{C_i}\leq t+\tau,\tau_{D_i}\neq\tau_{C_i}}\cdot\exp(-\sum_{s=0}^{\tau_{C_i}-t-2} f_{it}(s)\Delta t)\cdot\exp(-f_{it}(\tau_{C_i}-t-1)\Delta t) \\ & + \mathbf{1}_{t_{0_i}>t} + \mathbf{1}_{\tau_{C_i}\leq t}, \end{aligned} \tag{A.8}$$

$$\begin{aligned}\mathscr{L}_{\tau,i,t}(\mu;\tau_{C_i},\tau_{D_i},X_{it}) &= \mathbf{1}_{t_{0_i}\leq t,\tau_{C_i}>t+\tau+1}\cdot\exp(-\sum_{s=0}^{\tau}h_{it}(s)\Delta t) \\ &+ \mathbf{1}_{t_{0_i}\leq t,\tau_{D_i}=\tau_{C_i}\leq t+\tau}\cdot\exp(-\sum_{s=0}^{\tau_{C_i}-t-2}h_{it}(s)\Delta t) \\ &+ \mathbf{1}_{t_{0_i}\leq t,\tau_{C_i}\leq t+\tau,\tau_{D_i}\neq\tau_{C_i}}\cdot\exp(-\sum_{s=0}^{\tau_{C_i}-t-2}h_{it}(s)\Delta t)\cdot(1-\exp(-h_{it}(\tau_{C_i}-t-1)) \\ &+ \mathbf{1}_{t_{0_i}>t}+\mathbf{1}_{\tau_{C_i}\leq t}.\end{aligned} \tag{A.9}$$

Similar to previous literature, we can still decompose the likelihoods into terms involving different horizons of prediction τ. For each τ, we can consider as constant all terms involving previous horizons forward intensities. After log-linearization, the second decomposition is the following :

$$\begin{aligned}\mathscr{L}_{i,t}(\lambda;\tau_{C_i},\tau_{D_i},X_{it}) &= \mathbf{1}_{t_{0_i}\leq t,\tau_{C_i}>t+\tau+1}\cdot(-f_{it}(s)\Delta t)) \\ &+ \mathbf{1}_{t_{0_i}\leq t,\tau_{D_i}=\tau_{C_i}\leq t+\tau}\cdot\ln(1-\exp(-f_{it}(s)\Delta t)) \\ &+ \mathbf{1}_{t_{0_i}\leq t,\tau_{C_i}\leq t+\tau,\tau_{D_i}\neq\tau_{C_i}}\cdot(-f_{it}(s)\Delta t),\end{aligned} \tag{A.10}$$

$$\begin{aligned}\mathscr{L}_{i,t}(\mu;\tau_{C_i},\tau_{D_i},X_{it}) &= \mathbf{1}_{t_{0_i}\leq t,\tau_{C_i}>t+\tau+1}\cdot(-h_{it}(s)\Delta t) \\ &+ \mathbf{1}_{t_{0_i}\leq t,\tau_{C_i}\leq t+\tau,\tau_{D_i}\neq\tau_{C_i}}\cdot\ln(1-\exp(-h_{it}(s)\Delta t)).\end{aligned} \tag{A.11}$$

For all horizon of prediction s. We are now left with many small likelihoods that we can maximize separately instead of the huge maximum likelihood A.3. Ultimately, we can design two neural networks NN_λ and NN_μ with outputs N_{it}^λ and N_{it}^μ respectively to maximize the two above loss functions.

A.1.4 Distance-to-Default estimation

Distance-to-default

Let V_T be the firm value at time T, L the amount of debt to be repaid and E_T the equity value at time T. In case of bankruptcy, debt holders receive money before shareholders. The payoff to shareholders is then given by $E_T = \max(V_T - L, 0)$. This payoff is the same as a call option E_T on the underlying V_T with the strike being L. Merton (1974) model considers that V_t is following a geometric Brownian motion :

$$dV_t = \mu V_t d_t + \sigma V_t dB_t.$$

We can then use Black-Scholes formula to get the option price and firm's equity value E_t at any time t :

$$E_t = V_t N(d_t) - e^{-r(T-t)} \cdot L \cdot N(d_t - \sigma\sqrt{T-t}), \tag{A.12}$$

$$d_t = \frac{\ln(V_t/L) + (r + \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}},$$

with r being the risk-free rate and N(x) the normal cumulative distribution function. The distance-to-default is defined as the difference between the expected value of the asset and the default point. After substitution into a normal cumulative distribution function, we get :

$$DtD_t = \frac{\ln(\frac{V_A}{L}) + (\mu - \frac{\sigma_A^2}{2})(T-t)}{\sigma_A\sqrt{T-t}}. \tag{A.13}$$

Note that similarly to the variance restriction method, μ cannot be estimated precisely. Thus, we compute DtD with the following formula :

$$DtD = \frac{\ln(\frac{V_A}{L})}{\sigma_A\sqrt{T-t}}. \tag{A.14}$$

In Figure A.1 taken from Crosbie and Bohn (2003), the distance-to-default is denoted by "DD". The higher the asset value at horizon H the greater the distance-to-default will get, which lowers the default probability since the firm is more likely to be able to repay the debt owed. Similarly, the more debt the firm has, the smaller the DtD will be for a given level of asset value.

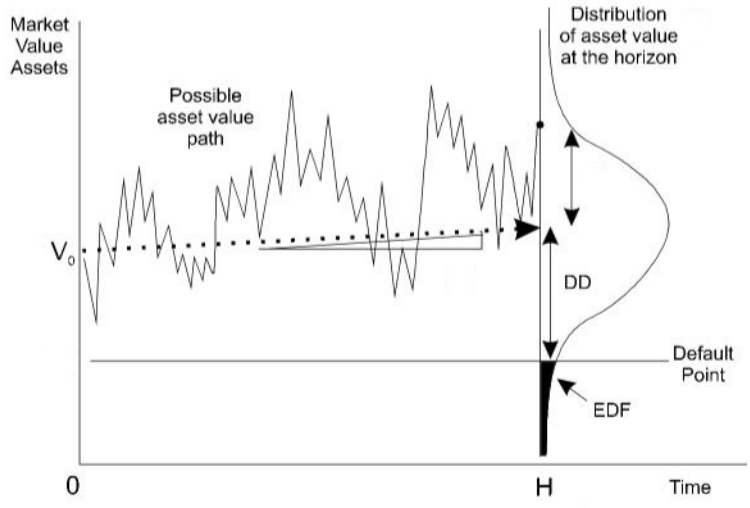


Figure A.1: Distance-to-default
The distance-to-default is denoted by DD. It is defined as the difference between the expected value of the asset and the default point.

Variance restriction method to estimate DtD

The variance restriction method is used in Duffie et al. (2007) to estimate the distance-to-default (DtD). This method is based on Merton (1974) model which states that the firm's equity value can be seen as a call option on the underlying asset and the strike being the amount of debt. This holds because stockholders receive money only once debt holders are fully paid. Applying the Black-Scholes call option formula to equity value, we get the following :

$$\begin{cases} V_E &= V_A N(d_1) - e^{-r(T-t)} \cdot D \cdot N(d_2) \\ d_1 &= \dfrac{\dfrac{V_A}{D} + (r - \dfrac{\sigma_A^2}{2}(T-t)}{\sigma_A \sqrt{T-t}} \\ d_2 &= d_1 - \sigma_A \sqrt{T-t}. \end{cases}$$

Where V_E is the market equity value, V_A is the market asset value, D is the default point and N the normal cumulative distribution function. Following KMV assumption, the default point D in the variance restriction method is specified as short-term debt plus one half of long-term debt.
Using Itô, we can show that

$$\sigma_E = \frac{V_A}{V_E} \cdot \frac{\partial V_E}{\partial V_A} \cdot \sigma_A.$$

Hence, the method consists of solving the following system with two equations and two unknowns V_A and σ_A :

$$\begin{cases} V_E &= V_A N(d_1) - e^{-r(T-t)} \cdot D \cdot N(d_2) \\ \sigma_E &= \dfrac{V_A}{V_E} \cdot \frac{\partial V_E}{\partial V_A} \cdot \sigma_A, \end{cases}$$

Once V_A and σ_A are estimated, the DtD is defined as the distance between the expected value of the asset and the default point. After substitution in a normal CDF, we get

$$DtD = \frac{\ln(\frac{V_A}{L}) + (\mu - \frac{\sigma_A^2}{2})(T-t)}{\sigma_A \sqrt{T-t}}. \tag{A.15}$$

However, many papers agree that μ is very tedious to estimate. Hence, DtD is often computed as

$$DtD = \frac{\ln(\frac{V_A}{L})}{\sigma_A \sqrt{T-t}}. \tag{A.16}$$

The major drawback of the variance restriction method used in Duffie et al. (2007) is the definition of the default point. The default point in the variance restriction method is following the so-called KMV assumption. This assumption states that for every firm the default point is exactly equal to short term debt plus one half of long-term debt. However, many financial firms do not account debt as short or long-term debt but as "other liabilities". This causes the

default point to be abnormally low for financial firms when using KMV assumption. Therefore, to take financial firms into account, we need to adjust the default point by taking into account other liabilities. The method proposed by Duan et al. (2012) employs a maximum likelihood to estimate the optimal fraction δ of other liabilities to include in the model.

Maximum likelihood estimation to estimate DtD

Duan et al. (2012) and Duan and Wang (2012) presented a method to estimate distance-to-defaults without having to exclude financial firms. The method accounts for other liabilities using a maximum likelihood estimation including a parameter δ to take into account other liabilities. The default point in this method becomes :

$$L = \text{short-term debt} + 0.5 \times \text{long-term debt} + \delta \times \text{other liabilities.} \tag{A.17}$$

Duan et al. (2012) and Duan and Wang (2012) usually estimate δ for many firms altogether (i.e. δ for a whole industry). However, we improve the methodology by computing δ for each firm individually. We should obtain higher deltas for financial firms than for non-financial firms. Estimating δ for each firm is highly time consuming because of greater computation time but it should highly improve the granularity and precision of the model. The log-likelihood function is given in Duan et al. (2012) and Duan and Wang (2012) by :

$$\begin{aligned}\mathscr{L}_i(\mu,\sigma,\delta) = &-\frac{n-1}{2}\ln(2\pi) - \frac{1}{2}\sum_{t=2}^{n}\ln(\sigma^2 h_t) - \sum_{t=2}^{n}\ln(\frac{\hat{V}_t(\sigma,\delta)}{A_t}) \\ &- \sum_{t=2}^{n}\ln(N(\hat{d}_t(\sigma,\delta))) - \sum_{t=2}^{n}\frac{1}{2\sigma^2 h_t} \times (\ln(\frac{\hat{V}_t(\sigma,\delta)}{\hat{V}_{t-1}(\sigma,\delta)} \cdot \frac{A_{t-1}}{A_t} - (\mu - \frac{\sigma^2}{2}))^2.\end{aligned}$$

where n is the number of period observations for each firm i. The likelihood above differs from Duan et al. (2012) and Duan and Wang (2012) because of the index i since we estimate the likelihood for each of the 2099 firms in our sample to get 2099 estimations of δ. Using this kind of likelihood is complex and very time consuming because we have to solve many inverse Black-Scholes formulas to get the time values of implied asset value $\hat{V}_t(\sigma,\delta)$ for each firm by solving equation A.12. However, inverse Black-Scholes formula does not have any closed form solution. The optimization is even more tedious since the implied asset value $\hat{V}_t(\sigma,\delta)$ depends on the final output of the likelihood δ. A_t is the book asset value and h_t is the time interval as a fraction of a year between two observations. h_t in the model is set to be 0.25 since we performed a linear interpolation when we had a missing value in the sample (see “Missing information"). To get rid of the inverse Black-Scholes formula problem, I use a dichotomic algorithm to compute the time series of asset values.

A.2 Appendix to Chapter 2

A.2.1 Technical proofs

Proof of Lemma 1

From the definition of DIG, we know that for any $R_k \notin \mathscr{PA}_j$, $I(R_k \to R_j || \mathscr{R}_{-\{k,j\}}) = 0$. This implies that for all t,

$$p(R_{j,t}|\mathscr{R}^{t-1}) = p(R_{j,t}|\mathscr{R}^{t-1}_{-\{k\}}). \tag{A.18}$$

On the other hand, by the assumption of the Lemma, $R_i \notin \mathscr{PA}_j$, we have

$$I(R_i \to R_j || \mathscr{R}_{-\{i,j\}}) = 0, \tag{A.19}$$

or equivalently, for all t,

$$p(R_{j,t}|\mathscr{R}^{t-1}) = p(R_{j,t}|\mathscr{R}^{t-1}_{-\{i\}}). \tag{A.20}$$

Combining (A.18) and (A.20) imply that for any pair $\{R_i, R_k\}$ that are not in the parent set of R_j, we have

$$p(R_{j,t}|\mathscr{R}^{t-1}_{-\{k\}}) = p(R_{j,t}|\mathscr{R}^{t-1}_{-\{i\}}). \tag{A.21}$$

To prove the claim of this lemma, we use (A.21) to show that all the time series in $\mathscr{R}_{-\{i,j\}} \setminus \mathscr{C}$ can be removed from the conditioning in (A.19). Let $R_k \in \mathscr{R}_{-\{i,j\}} \setminus \mathscr{C}$, by multiplying the above equality with $p(R_i^{t-1}|\mathscr{R}^{t-1}_{-\{i,k\}})$ and marginalizing over R_i^{t-1}, we obtain

$$\begin{aligned}
&\int p(R_{j,t}|\mathscr{R}^{t-1}_{-\{k\}}) p(R_i^{t-1}|\mathscr{R}^{t-1}_{-\{i,k\}}) dR_i^{t-1} = p(R_{j,t}|\mathscr{R}^{t-1}_{-\{i,k\}}) \\
&= \int p(R_{j,t}|\mathscr{R}^{t-1}_{-\{i\}}) p(R_i^{t-1}|\mathscr{R}^{t-1}_{-\{i,k\}}) dR_i^{t-1} = p(R_{j,t}|\mathscr{R}^{t-1}_{-\{i\}}).
\end{aligned}$$

The above equalities and (A.20) imply that for all t,

$$p(R_{j,t}|\mathscr{R}^{t-1}) = p(R_{j,t}|\mathscr{R}^{t-1}_{-\{i,k\}}),$$

or equivalently,

$$I(R_i \to R_j || \mathscr{R}_{-\{i,j,k\}}) = 0.$$

By repeating the above procedure, we obtain

$$I(R_i \to R_j || \mathscr{C}) = 0.$$

Proof of Lemma 2

Consider the VAR model in (2.13). First, we assume that X_i has no influence on X_j, i.e., $I(X_i \to X_j || \mathscr{X}_{-\{i,j\}}) = 0$ or equivalently $a_{j,i} = 0$ and show that (2.16) holds. Given this assumption, we have that for all t,

$$p(X_{j,t}|\mathscr{X}_{-\{i\}}^{t-1}) = p(X_{j,t}|\mathscr{X}^{t-1}).$$

Using the equations in (2.13) and the assumption that $a_{j,i} = 0$, we obtain

$$\begin{aligned}
&p(X_{j,t}|\mathscr{X}^{t-1}) = p(N_{j,t} + \sum_k a_{j,k} X_{k,t-1}|\mathscr{X}^{t-1}) = p(N_{j,t} + \sum_{k\neq i} a_{j,k} X_{k,t-1}|\mathscr{X}^{t-1}) \\
&= p(N_{j,t} + \sum_{k\neq i} a_{j,k} X_{k,t-1}| \sum_{k\neq i} a_{j,k} X_{k,t-1}, X_i^{t-1}, X_j^{t-1}) \\
&= p(N_{j,t} + \sum_{k\neq i} a_{j,k} X_{k,t-1}| \sum_{k\neq i} a_{j,k} X_{k,t-1}, X_j^{t-1}).
\end{aligned}$$

Note that we could replace $\mathscr{X}^{t-1}$ by $\{\sum_{k\neq i} a_{j,k} X_{k,t-1}, X_i^{t-1}, X_j^{t-1}\}$ or $\{\sum_{k\neq i} a_{j,k} X_{k,t-1}, X_j^{t-1}\}$ in the above equations, because given either of them $\sum_{k\neq i} a_{j,k} X_{k,t-1}$ becomes a constant and independent of $N_{j,t}$. By defining $Q_{t-1} := \sum_{k\neq i} a_{j,k} X_{k,t-1}$, the above equations can be rewritten as follows

$$p(X_{j,t}|\mathscr{X}^{t-1}) = p(X_{j,t}|Q_{t-1}, X_i^{t-1}, X_j^{t-1}) = p(X_{j,t}|Q_{t-1}, X_j^{t-1}), \forall\, t,$$

or equivalently,

$$\mathbb{E}\left[\log \frac{p(X_{j,t}|Q_{t-1}, X_i^{t-1}, X_j^{t-1})}{p(X_{j,t}|Q_{t-1}, X_j^{t-1})}\right] = 0, \forall\, t.$$

Using the definition of DI, the above equalities can be written in terms of DI as follows

$$I(X_i \to X_j || Q) = 0.$$

On the other hand, we have

$$[a_{j,1}, ..., a_{j,i-1}, a_{j,i+1}, ..., a_{j,m}] = \arg\min_{\mathbf{w}\in\mathbb{R}^{m-1}} \mathbb{E}\left[||X_{j,t} - \mathbf{w}^T \mathbf{X}_{-\{i\},t-1}||_2^2\right] := \mathbf{u}_t.$$

where $\mathbf{X}_{-\{i\},t-1} := [X_{1,t-1}, ..., X_{i-1,t-1}, X_{i+1,t-1}, ..., X_{m,t-1}]^T$. This means that $Q_{t-1} = \mathbf{u}_t^T \mathbf{X}_{-\{i\},t-1}$.

Next, we show the reverse direction, i.e., we assume (2.16) holds, then we show $I(X_i \to X_j || \mathscr{X}_{-\{i,j\}}) = 0$. To do so, it suffices to show $a_{j,i} = 0$. Since (2.16) holds, we have

$$p(X_{j,t}|\mathbf{u}_t^T \mathbf{X}_{-\{i\},t-1}, X_i^{t-1}, X_j^{t-1}) = p(X_{j,t}|\mathbf{u}_t^T \mathbf{X}_{-\{i\},t-1}, X_j^{t-1}), \forall\, t.$$

Using the j-th equation of (2.13) and the above equalities, for any instances $(\mathbf{u}_t^T \mathbf{x}_{-\{i\},t-1}, x_i^{t-1}, x_j^{t-1})$

of $(\mathbf{u}_t^T \mathbf{X}_{-\{i\},t-1}, X_i^{t-1}, X_j^{t-1})$, we obtain $\forall t$,

$$\mathbb{E}\left[X_{j,t} | \mathbf{u}_t^T \mathbf{x}_{-\{i\},t-1}, x_i^{t-1}, x_j^{t-1}\right] = \mathbb{E}\left[X_{j,t} | \mathbf{u}_t^T \mathbf{x}_{-\{i\},t-1}, x_j^{t-1}\right],$$

which implies

$$\begin{aligned}
&\mathbb{E}\left[N_{j,t}\right] + \sum_{k \neq i} a_{j,k} x_{k,t-1} + a_{j,i} x_{i,t-1} = \\
&\mathbb{E}\left[N_{j,t}\right] + \sum_{k \neq i} a_{j,k} x_{k,t-1} + a_{j,i} \mathbb{E}\left[X_{i,t-1} | \mathbf{u}_t^T \mathbf{x}_{-\{i\},t-1}, x_j^{t-1}\right].
\end{aligned}$$

This simplifies to

$$a_{j,i} x_{i,t-1} = a_{j,i} \mathbb{E}\left[X_{i,t-1} | \mathbf{u}_t^T \mathbf{x}_{-\{i\},t-1}, x_j^{t-1}\right], \forall t.$$

This equation should hold for any $x_{i,t-1}$. This is only possible if $a_{j,i} = 0$.

Proof of Lemma 3

The proof is similar to the linear version and uses the fact that exogenous noises $\{\varepsilon_{j,t}\}$ are independent. More precisely, we have

$$p(X_{j,t} | \mathscr{X}^{t-1}) = p(F_j(\mathscr{X}^{t-1}) + \varepsilon_{j,t} | F_j(\mathscr{X}^{t-1})).$$

Since there is no influence from X_i to X_j, we can eliminate it from the conditioning and the argument of function F_j and obtain

$$p\big(F_j(\mathscr{X}^{t-1}) + \varepsilon_{j,t} | \mathscr{X}^{t-1}\big) = p\big(F_j(\mathscr{X}_{-\{i\}}^{t-1}) + \varepsilon_{j,t} | \mathscr{X}_{-\{i\}}^{t-1}\big).$$

On the other hand, because given either $\{F_j(\mathscr{X}_{-\{i\}}^{t-1}), X_i^{t-1}, X_j^{t-1}\}$ or $\{F_j(\mathscr{X}_{-\{i\}}^{t-1}), X_j^{t-1}\}$, the value of $F_j(\mathscr{X}_{-\{i\}}^{t-1})$ is no longer a random variable. Using this relationship and the fact that $\varepsilon_{j,t}$ is independent of $\mathscr{X}^{t-1}$, we obtain

$$p\big(F_j(\mathscr{X}_{-\{i\}}^{t-1}) + \varepsilon_{j,t} | F_j(\mathscr{X}_{-\{i\}}^{t-1}), X_i^{t-1}, X_j^{t-1}\big) = p\big(F_j(\mathscr{X}_{-\{i\}}^{t-1}) + \varepsilon_{j,t} | F_j(\mathscr{X}_{-\{i\}}^{t-1}), X_j^{t-1}\big).$$

By defining $Q_{t-1} := F_j(\mathscr{X}_{-\{i\}}^{t-1})$, the above equations can be rewritten in terms of DI as follows,

$$I(X_i \to X_j || Q) = 0.$$

To show the reverse, we need to prove that $I(X_i \to X_j || \mathscr{X}_{-\{i,j\}}) = 0$ if Equation (2.18) holds. Because $I(X_i \to X_j || Q) = 0$ and using Equation (2.17), for all t, we have

$$p\big(F_j(\mathscr{X}^{t-1}) + \varepsilon_{j,t} | Q_{t-1}, X_i^{t-1}, X_j^{t-1}\big) = p\big(F_j(\mathscr{X}^{t-1}) + \varepsilon_{j,t} | Q_{t-1}, X_j^{t-1}\big),$$

where $Q_{t-1} = F_j(\mathscr{X}_{-\{i\}}^{t-1})$. Note that the conditioning on the right-hand-side distribution is

independent of X_i^{t-1}. This implies that function F_j does not depend on X_i. Therefore, we can remove X_i^{t-1} from the argument of F_j, i.e.,

$$X_{j,t} = F_j(\mathscr{X}^{t-1}) + \varepsilon_{j,t} = F_j(\mathscr{X}_{-\{i\}}^{t-1}) + \varepsilon_{j,t},$$

which further implies

$$p(F_j(\mathscr{X}_{-\{i\}}^{t-1}) + \varepsilon_{j,t} | \mathscr{X}_{-\{i,j\}}^{t-1}, X_i^{t-1}, X_j^{t-1}) = p(F_j(\mathscr{X}_{-\{i\}}^{t-1}) + \varepsilon_{j,t} | \mathscr{X}_{-\{i,j\}}^{t-1}, X_j^{t-1}).$$

This is equivalent to

$$I(X_i \to X_j || \mathscr{X}_{-\{i,j\}}) = 0.$$

A.2.2 Koopman-based Lifting Method

Let $\mathbf{X}_t := \{X_{1,t}, ..., X_{m,t}\}$ denote a network of m time series such that

$$\dot{\mathbf{X}}_t = F(\mathbf{X}_t), \tag{A.22}$$

where the vector field $F(\mathbf{X}) = (F_1(\mathbf{X}), ..., F_m(\mathbf{X}))$ is of the form

$$F_j(\mathbf{X}) = \sum_{k=1}^{K} w_{j,k} h_k(\mathbf{X}). \tag{A.23}$$

In the above equation, $w_{j,k} \in \mathbb{R}$ are unknown weights and $\{h_k(\mathbf{X})\}$ denote a set of known library functions, e.g., monomials. Furthermore, let $\varphi^t(\mathbf{X}_0)$ denote the solution to (A.22) associated with the initial condition $\mathbf{X}_0$.

Now, suppose that we have N noisy observations $\{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_N, \mathbf{y}_N)\}$ of the system trajectory, where $\mathbf{x}_i$ is the initial point and $\mathbf{y}_i$ is the final point after T_s steps, i.e.,

$$\mathbf{y}_i - \epsilon_i = \varphi^{T_s}(\mathbf{x}_i - \varepsilon_i), \;\; i = 1, ..., N,$$

where ϵ_i and ε_i are the measurement noises. The goal is to estimate the weights $\{w_{j,k}\}$ using these observations and consequently infer the causal network among the time series. To do so, we use the Koopman approach Mauroy and Goncalves (2019) that lifts the observation space to another space in which the relationships are linear. More precisely, the steps are as follows:

- Select a set of M basis lifting functions $\{p_1(\mathbf{x}), ..., p_M(\mathbf{x})\}$, and lift the observations,

$$\mathbf{P}_x := \begin{pmatrix} p_1(\mathbf{x}_1) & \cdots & p_M(\mathbf{x}_1) \\ p_1(\mathbf{x}_2) & \cdots & p_M(\mathbf{x}_2) \\ \vdots & \ddots & \vdots \\ p_1(\mathbf{x}_N) & \cdots & p_M(\mathbf{x}_N) \end{pmatrix}, \mathbf{P}_y := \begin{pmatrix} p_1(\mathbf{y}_1) & \cdots & p_M(\mathbf{y}_1) \\ p_1(\mathbf{y}_2) & \cdots & p_M(\mathbf{y}_2) \\ \vdots & \ddots & \vdots \\ p_1(\mathbf{y}_N) & \cdots & p_M(\mathbf{y}_N) \end{pmatrix}. \tag{A.24}$$

- Identify the Koopman operator $\mathbf{L} := \frac{1}{T_s}\log(\mathbf{P}_x^{\dagger}\mathbf{P}_y)$, where $\mathbf{P}_x^{\dagger}$ denotes the pseudo-inverse of $\mathbf{P}_x$ and the function log denotes the (principal) matrix logarithm.

- Identify the weights using the following equations: $\widehat{w}_{k,j} := [\mathbf{L}]_{k,l}$, with l such that $p_l(\mathbf{x}) = x_j$, where $\mathbf{x} = (x_1, ..., x_m)$.

An alternative approach to obtain the weights is the dual lifting method which executes the following steps instead of the above last step. At first, it finds matrix $\widehat{\mathbf{F}}$ using the following equation,

$$\widehat{\mathbf{F}}_{N\times m} := \mathbf{L}_{N\times N}\begin{pmatrix}\mathbf{x}_1^T\\ \vdots \\ \mathbf{x}_N^T\end{pmatrix}_{N\times m}.$$

Next, it constructs

$$\mathbf{H}_x := \begin{pmatrix} p_1(\mathbf{x}_1) & \cdots & p_M(\mathbf{x}_1)\\ \vdots & \ddots & \vdots \\ p_1(\mathbf{x}_N) & \cdots & p_M(\mathbf{x}_N)\end{pmatrix}_{N\times M},$$

and for each j, solve the following regression problem to get the weights

$$\widehat{\mathbf{w}}_j := \arg\min_{\mathbf{w}\in\mathbb{R}^M} \left|\left|\mathbf{H}_x\mathbf{w} - \widehat{\mathbf{F}}_{:,j}\right|\right|_2^2 + \rho||\mathbf{w}||_1,$$

where $\widehat{\mathbf{F}}_{:,j}$ denotes the j-th column of matrix $\widehat{\mathbf{F}}$ and $\widehat{\mathbf{w}}_j = [\hat{w}_{j,1}, ..., \hat{w}_{j,M}]^T$.

A.2.3 Ideal portfolio

In this section, we show how the ideal portfolio is related to the coefficients of the linear system in (2.13). Recall the optimization problem in Lemma 2.

$$\mathbf{u}_t := \arg\min_{\mathbf{w}\in\mathbb{R}^{m-1}} \mathbb{E}\left[||X_{j,t} - \mathbf{w}^T\mathbf{X}_{-\{i\},t-1}||_2^2\right],$$
$$\mathbf{X}_{-\{i\},t-1} := [X_{1,t-1}, \cdots, X_{i-1,t-1}, X_{i+1,t-1}, \cdots, X_{m,t-1}]^T.$$

Consider the j-th Equation in (2.13), i.e.,

$$X_{j,t} = \sum_{k=1}^{m} a_{j,k}X_{k,t-1} + N_{j,t}.$$

If $a_{j,i} = 0$, by substituting the above equation into the optimization, we obtain

$$\begin{aligned}
&\min_{\mathbf{w}\in\mathbb{R}^{m-1}} \mathbb{E}\left[|| \sum_{k\neq i} (a_{j,k} - w_k) X_{k,t-1} + N_{j,t} ||_2^2 \right] = \min_{\mathbf{w}\in\mathbb{R}^{m-1}} \mathbb{E}\left[||N_{j,t}||_2^2 \right] \\
&+ \mathbb{E}\left[|| \sum_{k\neq i} (a_{j,k} - w_k) X_{k,t-1} ||_2^2 \right] + 2\mathbb{E}\left[|| \left(\sum_{k\neq i} (a_{j,k} - w_k) X_{k,t-1} \right) N_{j,t} ||_2^2 \right] \\
&= \min_{\mathbf{w}\in\mathbb{R}^{m-1}} \mathbb{E}\left[|| \sum_{k\neq i} (a_{j,k} - w_k) X_{k,t-1} ||_2^2 \right] + \mathbb{E}\left[||N_{j,t}||_2^2 \right]
\end{aligned}$$

The last equality is due to the fact $N_{j,t}$ is independent of $\{X_{k,t-1}\}$ and have zero mean. This implies that the solution is $w_k = a_{j,k}$ for $k \in \{1, ..., i-1, i+1, ..., m\}$.

A.3 Appendix to Chapter 3

A.3.1 Tutorial for StockTwits Messages Extraction

We use stock price data from CRSP/Compustat of all US and Canadian listed stocks from January 2010 to March 2020. From this dataset, we create the list of unique tickers for which we will extract messages. We will later be able to merge the two datasets using the date and ticker for every observation. We use the StockTwits Application Programming Interface (API) to download messages from StockTwits. One query on StockTwits API is called a JavaScript Object Notation (JSON) request. Every message on StockTwits has a unique identifier ("msg_id") posted by a user with a unique identifier ("user_id"). JSON requests allow to query the database by ticker (called "symbol method") or by user (called "user method"). We use the query by ticker. One query only outputs the latest 30 messages concerning that ticker. However, it is possible to set a parameter ("max") to output the latest 30 messages up to this particular message identifier. This parameter allows us to crawl the message history of a ticker by recursively changing the "max" parameter to the oldest message identifier in the query. To perform a JSON request for Apple (AAPL) up to the message identifier 30'000'000, simply enter the following URL in a browser : https://api.stocktwits.com/api/2/streams/symbol/AAPL.json?&max=30000000. The page we get looks unreadable but it has always the same structure : several pairs of keys and values. The structure of JSON can easily be interpreted by modern programming languages. We create a Python script to query the API and extract the message history of every ticker in the ticker list. We store the output of every JSON request in .txt files in dedicated ticker folders.

A.3.2 Message Count

A StockTwits message can refer to multiple tickers. Figure A.2 shows the histogram of the number of tickers tagged per message. As the vast majority of message includes only one ticker, we only show on this plot messages referring to more than one ticker. The maximum number of tickers per message amounts to 28 and corresponds to 11 messages in the sample. Many messages refer to several tickers and this creates duplicates in the database because we consider the same message for all tickers tagged in the message.

Left plot of Figure A.3 shows the number of messages with and without double counting. In our sample, the number of messages without double counting is 76 million, as opposed to 90 million messages with double counting. Right plot of Figure A.3 shows the ratio between the number of messages with double counting and the number of messages without double counting. Throughout this paper, we only refer to the number of messages with double counting.

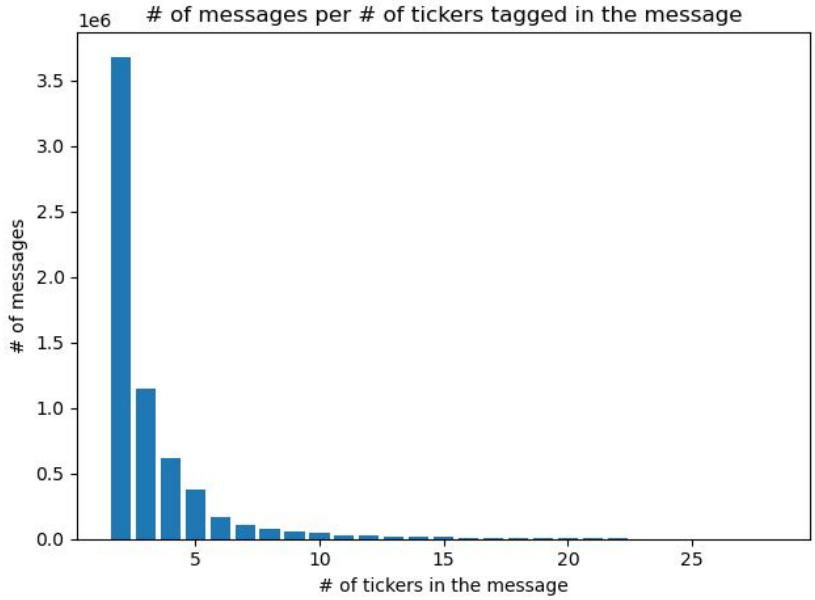


Figure A.2: Histogram of the number of tickers per message
Histogram of the number of tickers per message, across all messages referring to more than one ticker.

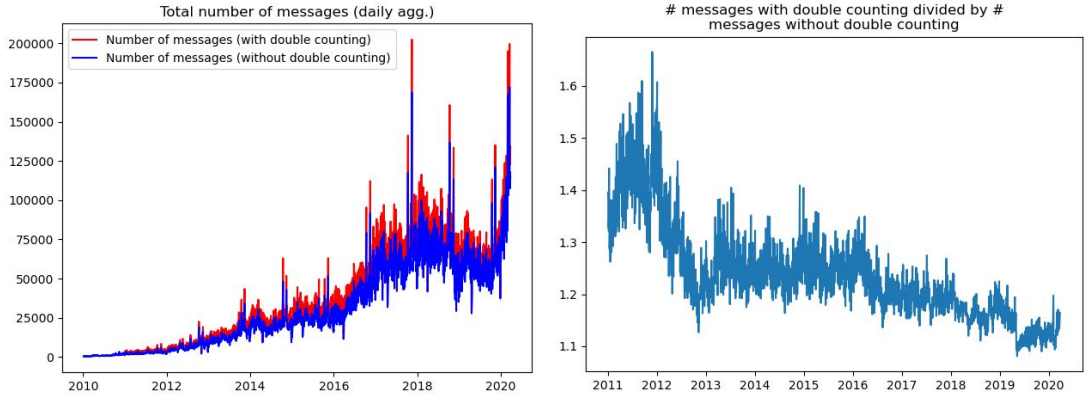


Figure A.3: Number of messages
Left plot shows the total number of messages with double counting (red) compared to the total number of messages without double counting (blue). Right plot shows the ratio between the number of messages with double counting and the number of messages without double counting. Numbers are aggregated daily.

A.3.3 User Summary Statistics

Left plot of Figure A.4 shows a log-log histogram of the number of followers per user and a right plot shows a log-log histogram of the number of messages posted by users. There are a few users with many followers (they can be seen as “influencers"), and many users with a few followers. In addition, most users seem to post on average between 10 and 400 messages and a few post a lot more. Overall, this appears to be a well balanced network structure. A more

detailed study of the network effects on market sentiment is beyond the scope of this paper.

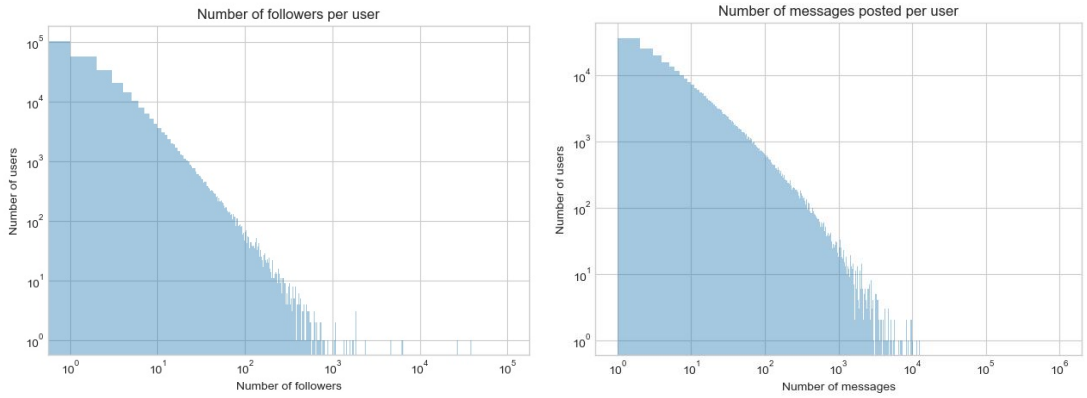


Figure A.4: User summary statistics
Left graph is a log-log histogram of the number of followers per user and the right graph shows the log-log histogram of the number of messages posted by users.

A.3.4 Anomalies

We discuss here two anomalies that appear in the word clouds in Figure 3.4.

The term “aldox” in the bullish cloud caught our attention. After some research, it is an abbreviation for Aldoxorubicin, a drug against tumors and is associated with pharmaceutical messages where investors were very enthusiastic about it. An example of a related message is “aldox is on the slide. have great faith this is truly world change". That is why the term is appearing almost exclusively in bullish messages, hence in the bullish cloud.

The bearish cloud contains the term "long position open", which seems like a bullish signal. Closer inspection shows that this term frequently appears in bearish user-labeled messages of intraday alerts such as “sell $labd close labd long position. open labd short position. time: 14:53 ny price: $13.64 zquant intraday alerts”. However, this anomaly is not an issue. We tested what happened when "long position open" is fed as a message into our sentiment classifier. As a message, it consists of the trigram "long position open", the two bigrams “long position” and “position open”, and the single words as unigrams. This results in a bullish score of 0.91 and the message is—correctly—classified as bullish.

A.3.5 Coverage

Stocktwits is neither regulated nor moderated, so one needs to filter the information that we use. Even if Stocktwits has valuable information from respected contributors, a blog[1] describes the concerns that may rise when using Stocktwits as a financial information provider, namely self-promotion, lack of credibility and other noise. To diversify noise and better extract

[1] https://www.warriortrading.com/stocktwits-review, last accessed on 1st of July 2022.

information, we exclude from our sample tickers that are rarely discussed. Thereto, we compute the median of daily message volume for each ticker and exclude from our sample tickers with a median of less than 50. Decreasing the median threshold increases the coverage at the expense of more noise in the daily polarity. Figure A.5 shows the coverage as a function of the median threshold. To increase the coverage we need to decrease the threshold a lot (e.g., decreasing the median threshold to 40 from 50 would increase the number of tickers covered to merely 22 from 19). We chose a median threshold of 50 as a balanced trade-off between noise and coverage.

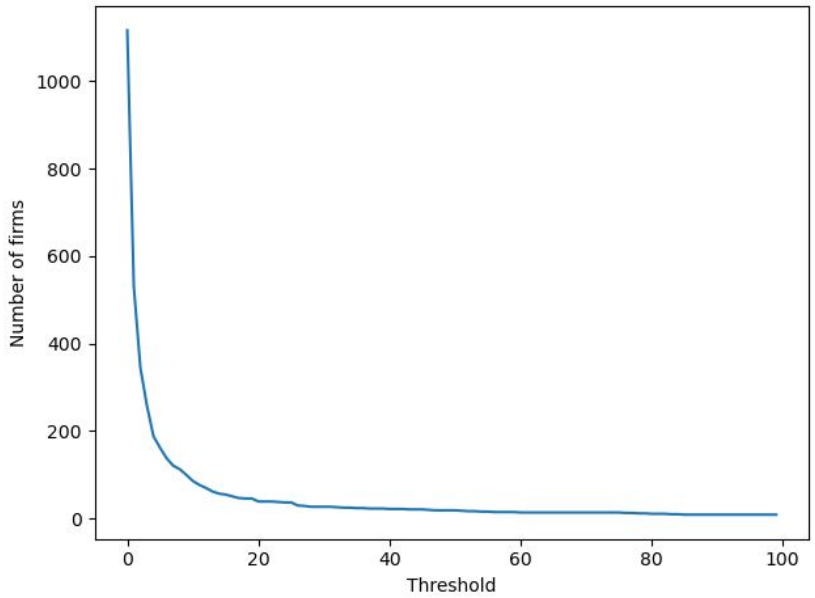


Figure A.5: Ticker coverage
Coverage as a function of the median threshold. A lower threshold increases the coverage at the expense of a bigger bias in the polarity.

Table A.1 shows the list of the 19 tickers above this threshold and their associated market capitalization as of 31st of December 2019. It appears that these most discussed tickers cover all sizes of stock, and hence we avoid big-firm bias. Also, it includes not only single firms but also ETFs on alternative investments. Finally, we cover several sectors so even if we have a restrictive universe, it is well diversified.

Ticker	Name	Market capitalization
AAPL	Apple	1287
AMD	Advanced Micro Devices	53
AMRN	Amarin	7
AMZN	Amazon	920
BABA	Alibaba	571
BAC	Bank of America	311
BB	BlackBerry	4
FB	Facebook	585
GLD	Gold ETF	59
IWM	Small-Cap ETF	55
JNUG	Direxion	0.5
MNKD	MannKind Corporation	0.2
NFLX	Netflix	142
PLUG	Plug Power	1
QQQ	Nasdaq100 ETF	134
SPY	S&P500 ETF	391
TSLA	Tesla	76
TWTR	Twitter	25
UVXY	VIX ETF	0.8

Table A.1: Ticker coverage
Coverage after the trimming process. List of tickers and corresponding market capitalization as of 31st of December 2019.

A.3.6 Classifier Performance

We first recap the definition of the basic performance measures for a binary classifier. First, one has to choose one class as the positive class. Instances (messages) are then divided according to their predicted and actual labels into true positives TP (predicted positive, actual positive), false positives FP (predicted positive, actual negative), true negatives TN (predicted negative, actual negative), and false negatives FN (predicted negative, actual positive). Precision $PRE = \frac{TP}{TP+FP}$ is the proportion of true positives among the predicted positives. Recall $REC = \frac{TP}{TP+FN}$ is the proportion of true positives among the actual positives. The precision-recall trade-off is captured by the F1 score, $\frac{2 \cdot PRE \cdot REC}{PRE+REC}$, the harmonic mean of precision and recall.

Tables A.2 and A.3 show the confusion matrices of our combined classifier out-of-sample and in-sample, respectively. We define accuracy as the fraction of correct predictions, omitting the messages with a predicted neutral sentiment. We thus obtain an out-of-sample accuracy of 85.9%. The in-sample accuracy is 87.4%.

		True	
		Bullish	Bearish
Predicted	Bullish	3'555'896	155'280
	Neutral	663'541	160'423
	Bearish	550'928	746'261

Table A.2: Confusion matrix for the combined classifier out-of-sample
Rows are the predicted class of the combined classifier, columns are the user-labels. Values are the number of messages in the corresponding category.

		True	
		Bullish	Bearish
Predicted	Bullish	14'433'375	532'509
	Neutral	2'656'730	642'329
	Bearish	1'992'535	3'071'837

Table A.3: Confusion matrix for the combined classifier in-sample
Rows are the predicted class of the combined classifier, columns are the user-labels. Values are the number of messages in the corresponding category.

A.3.7 Examples of Classified Messages

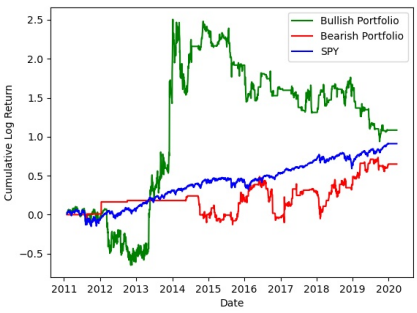
Here are some representative examples of classifications. Typical messages classified as bullish contain terms such as "buy buy" or "hope the pump come soon". Whereas typical bearish messages contain terms such as "sell everything" or "start short position here". Neutral

messages are either empty, or irrelevant to finance (e.g., "political posturing friend"[2]), or ambiguous (e.g., "lol wow").

A.3.8 Sentiment-Sorted Portfolios for Various Thresholds

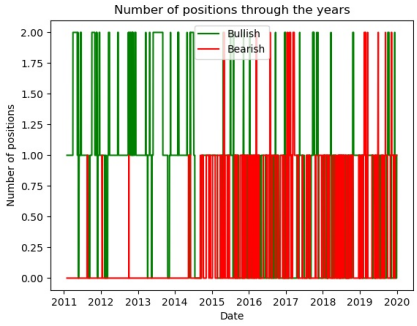
As the thresholds $U_t(x)$ and $L_t(x)$ are functions of the hyperparameter x, we provide for robustness check the results of our CAP-sorted portfolios for the values $x = 1.96$ (95% confidence band) in Figure A.6, and for $x = 2.81$ (99.5% confidence band) in Figure A.7. The portfolio performance arguably depends on the choice of x. In particular, the smaller x the more likely the bearish portfolio exhibits positive returns. On the other hand, the larger x the more likely the bullish portfolio misses the opportunities of positive returns. A careful gauging of x, possibly asymmetric in bullish and bearish, is therefore required for a real-world implementation of these strategies. Results could also improve for a larger cross-section of stocks than the 19 of our reduced sample.

[2] This is a reply to the message "honestly, how dumb can you be to believe that china was going to buy significant amount of agricultural products after the breakdown in trade talks. Even if they buy it will be just a little bit and not significant"

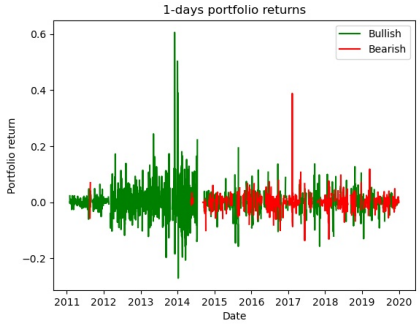


Cumulative log returns

Distribution of number of positions

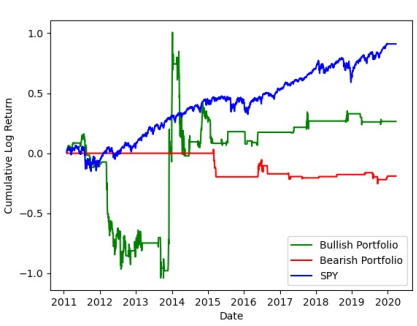


Number of positions over time

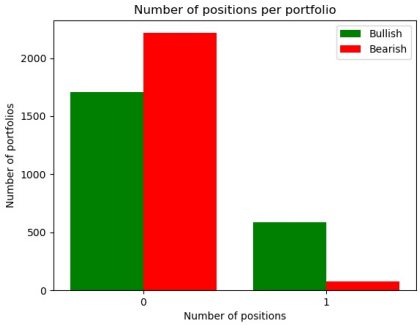


Daily returns

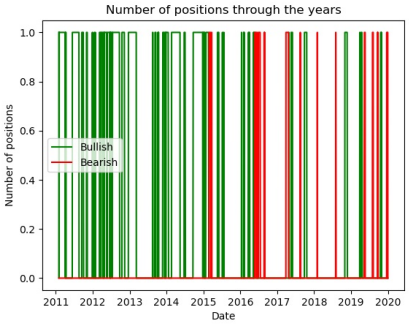
Figure A.6: Bullish and bearish portfolios for $x = 1.96$
Top left plot shows the cumulative log returns of the portfolios over the years, top right plot shows the distribution of the number of positions in the portfolios, bottom left plot is the number of positions over time and bottom right plot is the daily returns of both portfolios.



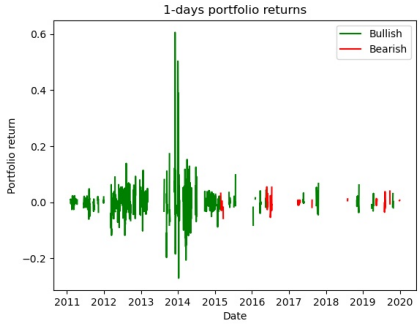
Cumulative log returns



Distribution of number of positions



Number of positions over time



Daily returns

Figure A.7: Bullish and bearish portfolios for $x = 2.81$
Top left plot shows the cumulative log returns of the portfolios over the years, top right plot shows the distribution of the number of positions in the portfolios, bottom left plot is the number of positions over time and bottom right plot is the daily returns of both portfolios.

Bibliography

Allen, F., Babus, A., and Carletti, E. (2010). Financial connections and systemic risk. *National Bureau of Economic Research*.

Altinbas, H. and Biskin, O. T. (2015). Selecting macroeconomic influencers on stock markets by using feature selection algorithms. *Procedia Economics and Finance*.

Altman, E., Kimura, H., and Barboza, F. (2017). Machine learning models and bankruptcy prediction. *Expert Systems with Applications*.

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*.

Antweiler, W. and Frank, M. Z. (2004). Is all that talk just noise? The information content of Internet stock message boards. *The Journal of Finance*.

Barigozzi, M. and Hallin, M. (2016). A network analysis of the volatility of high dimensional financial series. *Journal of the Royal Statistical Society*.

Bernardi, M. and Costola, M. (2019). High-dimensional sparse financial networks through a regularised regression model. *SAFE Working Paper*.

Bianchi, D., Billio, M., Casarin, R., and Guidolin, M. (2019). Modeling systemic risk with Markov switching graphical SUR models. *Journal of Econometrics*.

Billio, M., Casarin, R., and Rossini, L. (2019). Bayesian nonparametric sparse VAR models. *Journal of Econometrics*.

Billio, M., Getmansky, M., Lo, A. W., and Pelizzon, L. (2010). Measuring systemic risk in the finance and insurance sectors. *MIT Sloan School of Management Working Paper*.

Billio, M., Getmansky, M., Lo, A. W., and Pelizzon, L. (2012). Econometric measures of connectedness and systemic risk in the finance and insurance sectors. *Journal of Financial Economics*.

Bonaccolto, G., Caporin, M., and Panzica, R. (2019). Estimation and model-based combination of causality networks among large US banks and insurance companies. *Journal of Empirical Finance*.

Chawla, N. V., Japkowicz, N., and Kotcz, A. (2004). Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*.

Chen, E., Lu, Z., Xu, H., Cao, L., Zhang, Y., and Fan, J. (2020). A large scale speech sentiment corpus. *Proceedings of the 12th Language Resources and Evaluation Conference*.

Corsi, F., Lillo, F., Pirino, D., and Trapin, L. (2018). Measuring the propagation of financial distress with Granger-causality tail risk networks. *Journal of Financial Stability*.

Cover, T. M. and Thomas, J. A. (2012). Elements of information theory. *John Wiley & Sons*.

Crosbie, P. and Bohn, J. (2003). Modeling default risk. *Moody's KMV*.

Culver, W. J. (1966). On the existence and uniqueness of the real logarithm of a matrix. *Proceedings of the American Mathematical Society*.

Dekking, F., Kraaikamp, C., Lopuhaa, H., and Meester, L. (2005). A modern introduction to probability and statistics. *Springer*.

Diebold, F. X. and Yılmaz, K. (2014). On the network topology of variance decompositions: Measuring the connectedness of financial firms. *Journal of Econometrics*.

Duan, J.-C., Sun, J., and Wang, T. (2012). Multiperiod corporate default prediction — a forward intensity approach. *Journal of Econometrics*.

Duan, J.-C. and Wang, T. (2012). Measuring distance-to-default for financial and non-financial firms. *Global Credit Review*.

Duffie, D., Saita, L., and Wang, K. (2007). Multi-period corporate default prediction with stochastic covariates. *Journal of Econometrics*.

Erdemlioglu, D., Gillet, R. L., and Renault, T. (2017). Market reaction to news and investor attention in real time. *SSRN*.

Etesami, J., Habibnia, A., and Kiyavash, N. (2018). Econometric modeling of systemic risk: A time series approach. *SSRN*.

Etesami, J. and Kiyavash, N. (2014). Directed information graphs: A generalization of linear dynamical graphs. *American Control Conference*.

Fama, E. F. (1991). Efficient capital markets. *The Journal of Finance*.

Fama, E. F., Fisher, L., Jensen, M. C., and Roll, R. (1969). The adjustment of stock prices to new information. *International Economic Review*.

Ghoshal, S. and Roberts, S. (2016). Extracting predictive information from heterogeneous data streams using Gaussian processes. *Algorithmic Finance*.

Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*.

Hong, Y., Liu, Y., and Wang, S. (2009). Granger causality in risk and detection of extreme risk spillover between financial markets. *Journal of Econometrics*.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*.

Huang, C.-L. and Wang, C.-J. (2006). A GA-based feature selection and parameters optimization for support vector machines. *Expert Systems with Applications*.

Iacopini, M. and Rossini, L. (2019). Bayesian nonparametric graphical models for time-varying parameters VAR. *SSRN*.

Jiao, J., Permuter, H. H., Zhao, L., Kim, Y.-H., and Weissman, T. (2013). Universal estimation of directed information. *Information Theory*.

Kalli, M. and Griffin, J. E. (2018). Bayesian nonparametric vector autoregressive models. *Journal of Econometrics*.

Ke, Z., Kelly, B. T., and Xiu, D. (2020). Predicting returns with text data. *SSRN*.

Kim, S., Putrino, D., Ghosh, S., and Brown, E. N. (2011). A Granger causality measure for point process models of ensemble neural spiking activity. *PLoS one*.

Koopman, B. O. (1931). Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences of the United States of America*.

Leippold, M., Maire, B., and Blochlinger, A. (2012). Are ratings the worst form of credit assessment apart from all the others? *Swiss Finance Institute Research Paper*.

Longin, F. and Solnik, B. (2001). Extreme correlation of international equity markets. *The Journal of Finance*.

Lorenz, M. O. (1905). Methods of measuring the concentration of wealth. *Publications of the American Statistical Association*.

Loughran, T. and McDonald, B. (2011a). Barron's red flags: do they actually work? *Journal of Behavioral Finance*.

Loughran, T. and McDonald, B. (2011b). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of Finance*.

MacKinlay, A. C. (1997). Event studies in economics and finance. *Journal of Economic Literature*.

Mann, H. B. and Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*.

Massey, J. (1990). Causality, feedback and directed information. *International Symposium on Information Theory*.

Mauroy, A. and Goncalves, J. (2019). Koopman-based lifting techniques for nonlinear systems identification. *IEEE Transactions on Automatic Control*.

Merton, R. C. (1974). On the pricing of corporate debt: the risk structure of interest rates. *The Journal of Finance*.

Noshad, M., Zeng, Y., and Hero, A. O. (2019). Scalable mutual information estimation using dependence graphs. *International Conference on Acoustics, Speech and Signal Processing*.

Paninski, L. (2003). Estimation of entropy and mutual information. *Neural Computation*.

Petrova, K. (2019). A quasi-Bayesian local likelihood approach to time-varying parameter VAR models. *Journal of Econometrics*.

Piramuthu, S. (2004). Evaluating feature selection methods for learning in data mining applications. *European Journal of Operational Research*.

Psaradakis, Z., Ravn, M. O., and Sola, M. (2005). Markov switching causality and the money–output relationship. *Journal of Applied Econometrics*.

Qasem, M., Thulasiram, R., and Thulasiram, P. (2015). Twitter sentiment classification using machine learning techniques for stock markets. *International Conference on Advances in Computing, Communications and Informatics*.

Quinn, C., Kiyavash, N., and Coleman, T. P. (2015). Directed information graphs. *Transactions on Information Theory*.

Quinn, C. J., Coleman, T. P., Kiyavash, N., and Hatsopoulos, N. G. (2011). Estimating the directed information to infer causal relationships in ensemble neural spike train recordings. *Journal of Computational Neuroscience*.

Quinn, C. J., Kiyavash, N., and Coleman, T. P. (2013). Efficient methods to compute optimal tree approximations of directed information graphs. *IEEE Transactions on Signal Processing*.

Quinn, C. J., Pinar, A., and Kiyavash, N. (2017). Bounded degree approximations of stochastic networks. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*.

Ranco, G., Aleksovski, D., Caldarelli, G., Grčar, M., and Mozetič, I. (2015). The effects of Twitter sentiment on stock price returns. *PloS one*.

Renault, T. (2017). Intraday online investor sentiment and return patterns in the US stock market. *Journal of Banking & Finance*.

Renault, T. (2020). Sentiment analysis and machine learning in finance: a comparison of methods and models on one million messages. *Digital Finance*.

Saif, H., Fernández, M., He, Y., and Alani, H. (2014). On stopwords, filtering and data sparsity for sentiment analysis of Twitter. *International Conference on Language Resources and Evaluation*.

Schönbucher, P. (2003). Credit derivatives pricing models. *Wiley*.

Sheskin, D. J. (1998). Handbook of parametric and nonparametric statistical procedures. *Chapman and Hall*.

Sprenger, T. O., Tumasjan, A., Sandner, P. G., and Welpe, I. M. (2014). Tweets and trades: the information content of stock microblogs. *European Financial Management*.

Sricharan, K., Raich, R., and Hero, A. O. (2011). K-nearest neighbor estimation of entropies with confidence. *Information Theory Proceedings*.

Tetlock, P. (2007). Giving content to investor sentiment: the role of media in the stock market. *The Journal of Finance*.

Tripathy, R. K. and Bilionis, I. (2018). Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification. *Journal of Computational Physics*.

Tsai, C.-F. (2009). Feature selection in bankruptcy prediction. *Knowledge-Based Systems*.

Tukey, J. W. (1977). Exploratory data analysis. *Reading*.

Ullah, I. and Petrosino, A. (2016). About pyramid structure in convolutional neural networks. *International joint conference on neural networks*.

Wiener, N. (1956). The theory of prediction. *Modern Mathematics for Engineers*.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. *Conference on Empirical Methods in Natural Language Processing*.

Yildirim, S., Jothimani, D., Kavaklioglu, C., and Basar, A. (2018). Classification of hot news for financial forecast using NLP techniques. *IEEE International Conference on Big Data*.

Yuqinq, H., Kamaladdin, F., and Lipo, W. (2013). Feature selection for stock market analysis. *International Conference on Neural Information Processing*.

Divernois Marc-Aurèle

Rue du Midi 34, CH-1800 Vevey
27.02.1992, Swiss nationality
Email: divernois@gmail.com
Mobile: +41 79 756 47 53
Git: marcaureledivernois

Education

EPFL — Lausanne, CH
Ph.D., Advisor: Damir Filipovic — 2017–Current
- Area of research: "Machine Learning applied to Risk Management."

HEC Lausanne — Lausanne, CH
M.S. in Finance, GPA: 5.60/6.00 — 2013–2015
- Thesis: "Estimation of a forward intensity model for corporate default prediction".

HEC Lausanne — Lausanne, CH
B.S. in Economics, GPA: 5.30/6.00, ranked 6th in graduating class. — 2010–2013

Experience

Lombard Odier Asset Management — Geneva, CH
Risk Analyst — 2015–2017
- Responsible for the daily monitoring of performance and risk of >200 Multi-Asset and Fixed Income portfolios and mandates. Analyzed the risk metrics and exposures for key portfolios to track any material changes in the portfolio positioning.

Lombard Odier Asset Management — Geneva, CH
Analyst — 2015
- Creation of a VBA platform in order to automate all the reporting of the fund (CHF 7.5 billion AuM).

Northlight Group — London, UK
Intern — Summer 2014
- Analyst for the investment team of a credit hedge fund, participating in various tasks such as high yield loan modeling, equity research and firm valuation.

Technical skills

- **Data Science/Machine Learning with Python**
  - **Supervised & Unsupervised:** Regressions, PCA, Clustering, Numpy, Pandas, Scikit-Learn, Statsmodels.
  - **Web Data Scraping:** Requests, BeautifulSoup, Selenium, JSON.
  - **Natural Language Processing :** Sentiment Analysis, NLTK, SpaCy, TFIDF, Word embeddings.
  - **Deep Learning :** Artificial Neural Networks, CNN, RNN, TensorFlow, Keras, PyTorch.
- **Programming Languages**
  - Python, Matlab, R, SQL, VBA, Stata, HTML, CSS, Git, Office, LaTeX.

Publications

[1] M.-A. Divernois and D. Filipovic, "StockTwits Classified Sentiment and Stock Returns", 2022.

[2] M.-A. Divernois, J. Etesami, D. Filipovic, and N. Kiyavash, "Firm Networks Using Granger Causality", submitted to Journal of Econometrics, 2021.

[3] M.-A. Divernois, "A Deep Learning Approach to Estimate Forward Default Intensities", in *Swiss Finance Institute Research Paper No. 20-79*, 2019.

Conferences talks

- **Applied Machine Learning Days** 2022
  Speaker of the track of Advances of Machine Learning Approaches for Financial Decision Making & Time Series Analysis.
- **SIAM Conference on Financial Mathematics and Engineering** 2021
  Presented my paper "StockTwits Classified Sentiment and Stock Returns".
- **SFI Research Days** 2020
  Presented my paper "A Deep Learning Approach to Estimate Forward Default Intensities".
- **Swissquote Conference on Artificial Intelligence in Finance** 2019
  Speaker at the EPFL Finance and Technology Program.

Teaching

- **Head Teaching Assistant** at EPFL 2017-2022
  Financial Big Data (M.Sc. class, Prof. D. Challet) ∼50 students.
  Advanced Derivatives (M.Sc. class, Prof. E. Perazzi) ∼40 students.
- **Teaching Assistant** at HEC 2011-2014
  Economics I : microeconomics (B.Sc. class, Prof. T. von Ungern) ∼900 students.
  Economics II : macroeconomics (B.Sc. class, Prof. C. Sfreddo) ∼600 students.
  Principles of Finance (B.Sc. class, Prof M. Rockinger) ∼300 students.

Awards

- **Best Teaching Assistant 2021** 2021
  Elected Best Teaching Assistant by the students of the Master in Financial Engineering at EPFL. Prize awarded during the graduation ceremony.
- **Winner of Saxo Bank's portfolio management simulation** 2014
  Highest portfolio value out of ∼750 participants at the end of a six weeks stock market game held by SaxoBank.
- **Winner of HEC Business Game** 2013
  Team ranked first out of 64 participants on a real-time business strategy simulation held by several professors using ERPSim.

Languages

- **French:** Mother tongue.
- **English:** Fluent.
- **German:** Intermediate B1.

Extracurricular Activities

- **Chess club Teacher** 2018
  Giving chess lessons to children up to 1500ELO.
- **Head of Junior Entreprise** 2014
  Student club offering consulting services to the market.