Probabilistic Framework for Modeling Event Shocks to Financial Time Series

Probabilistic Framework for Modeling Event Shocks to Financial Time Series

Yada Zhu*‡
MIT-IBM Watson AI Lab
Cambridge, MA, USA
yzhu@us.ibm.com

Wenyu Chen†‡
Massachusetts Institute of Technology
Cambridge, MA, USA
wenyu@mit.edu

Yang Zhang‡
MIT-IBM Watson AI Lab
Cambridge, MA, USA
yang.zhang2@ibm.com

Tian Gao‡
IBM Research
Yorktown Heights, NY, USA
tgao@us.ibm.com

Jianbo Li‡
Three Bridges Capital
New York, NY, USA
jianboliru@gmail.com

ABSTRACT

In financial market, certain types of stochastic events are intrinsically impactful to the prediction of financial times series, such as stock return, while few existing research attempts have been made to incorporate stochastic event modeling to time series modeling in a principled way. In this paper, we present a pioneering study that fills this gap. In particular, we introduce a generic probabilistic model that captures 1) the inter-dependencies among stochastic events, and 2) the impact of these events on time series. To this end, we extend multivariate Hawkes process (MHP) and proximal graphical event model (PGEM) and apply this framework to modeling two financial events, companies' quarterly revenue releases and updates of consensus prediction of quarterly revenue, and their impacts on the mean and correlation structures of future stock return. Our model not only improves prediction of financial time series, but also promotes AI trust for finance by revealing the causal relationship among the events. Extensive experimental results based on real financial market data validate the effectiveness of our models in learning event impact and improving investment decision by incorporating stochastic event impacts.

CCS CONCEPTS

• **Mathematics of computing** → *Stochastic processes*; **Multivariate statistics**.

KEYWORDS

Hawkes process, graphical event, quarterly revenue, stock return, variance-covariance

*To whom correspondence should be addressed.
†W. Chen did the work while being an intern at IBM Research.
‡These authors had equal contribution.

ICAIF'21, November 3–5, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-9148-1/21/11…$15.00
https://doi.org/10.1145/3490354.3494407

ACM Reference Format:
Yada Zhu, Wenyu Chen, Yang Zhang, Tian Gao, and Jianbo Li. 2021. Probabilistic Framework for Modeling Event Shocks to Financial Time Series. In *2nd ACM International Conference on AI in Finance (ICAIF'21), November 3–5, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3490354.3494407

1 INTRODUCTION

Time series is a collection of random variables observed sequentially at fixed intervals of time and it is of paramount interest in finance field and artificial intelligent (AI) area. Learning from time series provides valuable insights for market movement and future stock return and correlation that are essential for investment decision making. A prominent issue of financial time series analysis, especially for predicting future value of company stocks, is the dynamic impact of various events, such as release of quarterly earning reports, announcement of new products and change of credit rating, as well as the price chain reactions due to the correlations of event/time series entities [15]. In contrast to time series, event streams are sequences of events of various types that typically occur as irregular and asynchronous continuous-time arrivals. It is well known that event streams can be modeled using (multivariate) point process [1, 8]. Both time series and stochastic point process in general domains have been well studied and abundant methods have been proposed to deal with each of them independently, while few attempts have been made to incorporate the impact of stochastic events into time series modeling and forecasting in a principled way. The challenge lies in the sophisticated temporal interaction among different events and their shock effects to time series, which can also have temporal cross-correlations for multivariate time series. Despite recently point process has gained growing interests for finance applications, such as modeling the joint dynamics of trades and mid-price changes of the NYSE [3] and pricing options [18], few research attempts have been made to fill in this gap.

In this paper, we present a pioneering study to incorporate impact of stochastic events into multivariate time series modeling and demonstrate its application in capturing effects of crucial events on stock return and correlation prediction. Specifically, we consider two types of events, i.e., quarterly revenue release of public companies and updates of consensus estimate on the quarterly revenue. Revenue measures a company's earning power which is a key indicator of future stock returns. As time close to revenue release, more information about a company's financial status becomes available, and consensus is more frequently updated by analysts to incorporate the new information. Consequently, stock price changes immediately to reflect the latest market information. Therefore, to predict the upcoming consensus update and stock return, we propose a generic probabilistic framework to model event intensity, event magnitude, and their effects to the distribution of stock return. To this end, we develop a set of probabilistic models including extended multivariate Hawkes process (MHP) [17, 24] with different kernels and a novel proximal graphical event model (PGEM). The extended MHP models explicitly leverage domain knowledge on event interactions and impact of every historical events. PGEM learns historical impacts on events from a short window in the most recent past. It learns not only the density of event occurrence but also event causal relationship which sheds light on AI trust for finance. Further, we capture the impact of previous event magnitude to consensus magnitude by adjusting the distribution mean under liquidity market assumption. Following this, we incorporate the effects of the two events into stock return distribution parameters. This leads to time-varying mean and stock correlation structure which overcome the challenge of prediction stock returns due to stochastic event impact.

The main contributions of the paper are summarized as follows:

(1) propose a generic framework for capturing the impact of stochastic events to time series modeling;
(2) propose novel and expandable probabilistic models to learn the intensity of event occurrence and an explainable graphical representation of event causal relationship;
(3) propose approaches to learn dynamic stock return distributions and correlation structure under explicit impact of quarterly revenue release and consensus update;
(4) evaluate the performance of the proposed framework and variants of models with two sets of financial market data in terms of interpretations of event inter-dependency, effectiveness of event magnitude prediction, and profitability of simulated portfolios.

2 RELATED WORK

There is rich literature in time series analysis as well as event modeling via stochastic point process, e.g. [3, 13, 22, 25]. However, these works do not really combine the event models with the time series analysis, and apply them in the financial settings. To the best of our knowledge, this is the first work that combines both aspects, modeling the event occurrences via the point process and further predicting the future time series based on the event models.

One of the most related works that consider both time series and event modeling aspects is [21]—they model the event sequence based on the time series input via the combined effects of two recurrent neural networks. These two RNNs take time series and event sequences as input, respectively, and thus they are able to capture the information from regular time series, and handle the irregularity caused by the event occurrences at the same time. However, their loss function does not look at the likelihood of the event occurrences under the point process model, and they do not investigate into the structure of dependencies on the history events. Based on [21], [20] adopt the composition structure of the intensity function, and consider the maximum likelihood of the corresponding point process. Furthermore, they use an additional attention layer based on the infectivity matrix between different events to model the dependencies on the history events. However, attention does not necessarily indicates causation, and it is not straightforward to interpret attention over time. Both of [20, 21] consider learning the intensity function of event models based on the information of time series, but they do not consider the opposite as we do in this paper, i.e. the impact of the event models on the time series forecasting.

3 PROBLEM FORMULATION

Our general goal is to construct a probabilistic model that describes the inter-dependencies among some key financial variables and among different companies, so that it can predict the future values of these variables based on the historical observations. This section will formulate the problem and concretize the financial variable to be studies and modelel.

3.1 Time Series v.s. Event Variables

Before we introduce our financial variables of interest, we would like to emphasize that these variables can be categorized into two data types – time series and events.

A *time series variable*, $\boldsymbol{X}$, refers to a collection of random variables corresponding to *every time step*. Formally,

$$\boldsymbol{X} = \{X_{it}\},$$

where t represents the time step (whose temporally granularity is by day), and i represents the company. We let d denote the number of companies we consider and T denote the time horizon.

An *event variable*, $\boldsymbol{E}$, models financial variables that occur sporadically, *i.e.* do *not* have observations at all the timestamps. Therefore, each event is associated with two variables, the time variable, which depicts the timestamps when the event happens, and the magnitude variable, which depicts the magnitude of the event. Formally,

$$\boldsymbol{E} = \{l_{in}, E_{in}, t_{in}\},$$

where n denotes the event index, l_{in} is the label of the n-th event, (e.g. $l_{in} = z$ for company release and $l_{in} = c$ for consensus correction), E_{in} denotes the magnitude of the n-th event, and t_{in} denotes the timestamp of the n-th event.

3.2 Variables of Interest

Although the proposed framework is generic and can be applied to a wide variety of financial variables, for concreteness, we will focus on the following three financial variables:

• **Company Revenue** Company revenue, denoted as $\boldsymbol{Z}$, refers to the quarterly revenue released by each public company, which is a key indicator of the valuation and profitability of a company and used to project future stock returns by investors. $\boldsymbol{Z}$ is considered as an event variable (hence $\boldsymbol{Z} = \{z, Z_{in}, t^Z_{in}\}$) because usually a company releases its revenue once per quarter, whereas our temporal granularity is by day.

• **Consensus** Consensus, denoted as $\boldsymbol{C}$, refers to the aggregated revenue estimation of the upcoming quarters by legions of stock analysts. Consensus usually remains constant until it is adjusted sporadically, so it is considered as an event variable (hence

$C = \{c, C_{in}, t^C_{in}\}$). Note consensus change and consensus adjustment are used interchangeably with consensus update in this paper.

• **Stock Price and Return** Stock price, S, refers to the the daily closing stock price of a company. It is considered as a time series variable (*i.e.* $S = \{S_{it}\}$). Define stock return, $R = \{R_{it}\}$ as

$$R_{it} = \frac{S_{it}}{S_{i(t-1)}} - 1.$$

Since stock return exhibits nicer statistical properties than stock price, we will be using stock return in our model.

3.3 Objective

Given any timestamp t, let $\mathcal{H}(t)$ denote the historical observations of all the aforementioned variables up to time t. Our goal is to predict the upcoming consensus adjustment and stock return. Formally, our goal is to compute the following conditional probability densities given the history $\mathcal{H}(t)$

$$p\left(t^E_{i(n+1)}|\mathcal{H}(t)\right), p\left(E_{i(n+1)}|\mathcal{H}(t)\right), \text{ and } p\left(R_{i(t+1)}|\mathcal{H}(t)\right). \quad (1)$$

where, unless specified otherwise, n refers to the closest event index before t. The inferred values can then be made or simulated according to these distributions, *e.g.* by taking the expectation.

4 METHODOLOGY

We will solve the aforementioned problem using a probabilistic framework, which consists of two steps. First, we will construct a probabilistic model for the aforementioned three variables. Second, we will estimate the parameters of the probabilistic model.

4.1 Overview

Throughout this section, we denote $\mathcal{H}^Z(t)$, $\mathcal{H}^C(t)$, and $\mathcal{H}^R(t)$ as the historical earning magnitude, consensus magnitude, and stock returns, respectively. The construction of the probabilistic dependencies conforms to many financial intuitions and financial decision processes. Specifically, the time of the revenue release, t^Z_{in}, is exogenously given. The magnitude of the revenue release, Z_{in} depends on its own historic values, $\mathcal{H}^Z(t)$, and the consensus historical values, $\mathcal{H}^C(t)$. Then, the consensus update time, t^C_{in}, depends on the last and future revenue release times, *i.e.* $t^Z_{in'}$ and $t^Z_{i(n'+1)}$, where n' denotes the latest earning release event index before the current time t, as well as previous consensus update times, $t^C_{i(n-1)}, t^C_{i(n-2)}$, etc. The consensus adjustment value C_{in} depends on the previous consensus value $\mathcal{H}^C(t)$ and the historical revenue of a company $\mathcal{H}^Z(t)$. The stock return R_{it} depends on the historical returns $\mathcal{H}^R(t)$ and consensus $\mathcal{H}^C(t)$. To sum up, the conditional probabilities upon the historical variables can be expressed as

$$\begin{aligned} &p\left(Z_{in}|\mathcal{H}(t)\right) = p\left(Z_{in}|\mathcal{H}^Z(t), \mathcal{H}^C(t)\right),\\ &p\left(t^C_{in}|\mathcal{H}(t)\right) = p\left(t^C_{in}|t^Z_{in'}, t^Z_{i(n'+1)}, t^C_{i(n-1)}, t^C_{i(n-2)}, \dots\right),\\ &p\left(C_{in}|\mathcal{H}(t)\right) = p\left(C_{in}|\mathcal{H}^C(t), \mathcal{H}^Z(t)\right),\\ &p\left(R_{it}|\mathcal{H}(t)\right) = p\left(R_{it}|\mathcal{H}^R(t), \mathcal{H}^C(t)\right). \end{aligned} \quad (2)$$

The following subsections detail how each of the above probabilities is modeled.

4.2 Consensus Adjustment Time (t^C_{in}) Modeling

The distribution of t^C_{in} is parameterized by an event intensity parameter, $\lambda^C_i(t)$, which depicts the probability density of the event happening at time t. More formally, the probability of an event happening during the infinitesimal time interval $[t, t+dt]$ is given by $\lambda^C_i(t)dt$. According to Eq. (2), $\lambda^C_i(t)$ should be a function of the last and future revenue release times. We consider several ways of modeling $\lambda^C_i(t)$.

4.2.1 *Multivariate Hawkes process (MHP) with domain knowledge.*

The Hawkes processes [11] is known as a self-exciting process, where the occurrences of events will further increase the intensity of event happening. We use the multivariate Hawkes process (MHP, see e.g. [17, 24]), which extends this self-excitation to the mutual excitation of the events of different entities. In addition, based on both the domain knowledge and our observation on the data (see Fig. 2), the consensus updates tend to be more frequent when they are close to the revenue release dates due to more information available and increased interests of investors. Therefore, we make some adjustments to the intensity expression of our MHP model. To be more specific, given the history $\mathcal{H}(t)$, the intensity of $\lambda^C_i(t)$ company i is modeled as

$$\begin{aligned} \lambda^C_i(t) = \lambda_i &+ \sum_{j=1}^{d} \sum_{n:t^C_{jn}<t} \alpha_{i|j}\, g\left(t - t^C_{jn}; w_{i|j}\right)\\ &+ \alpha^+_i g\left(t^+_i - t; w^+_i\right) + \alpha^-_i g\left(t - t^-_i; w^-_i\right), \end{aligned} \quad (3)$$

where $g(\cdot; w)$ is called the nonnegative triggering kernel parametrized by w, and t^+_i and t^-_i denote the time of the next/last revenue release at time t, respectively. In this expression, the first two terms come from the original MHP model, while the last two terms account for the adjustment as discussed above. In the experiments, we consider two triggering kernels:

(1) exponential decay kernel,

$$g(t; w) = \exp(-t/w)/w;$$

(2) sigmoid kernel,

$$g(t; w) = \frac{e^w}{e^t + e^w}.$$

We learn the model through the maximum likelihood estimator, where the log-likelihood for the model is given by

$$\log L(\lambda, \alpha, w) = \sum_{i=1}^{d} \sum_{n=1}^{N^C_i} \log \lambda^C_i\left(t^C_{in}\right) - \sum_{i=1}^{d} \int_0^T \lambda^C_i(t)\, dt, \quad (4)$$

where N^C_i denotes the number of consensus events for company i. Following [17, 24], we penalize $\alpha_{i|j}$ with ℓ_1 regularizer to impose the sparsity.

4.2.2 *Proximal Graphical Event Model (PGEM).*

Typical Hawkes processes assume every historical release would impact new consensus updates, which contradicts the intuition that only the most recent few revenue reports would be impactful in practice. Due to such a long historical influence, it is also very hard to interpret the learned process to pin-point the exact factors that most influence new consensus updates. Therefore, we propose a different model, where the intensity model of λ_i^C is defined using the proximal assumption, which states that historical impacts on events only come from a short window in the most recent past. This assumption results in a proximal point process model (PGEM), such as the graphical event model [5]. A PGEM $\mathcal{M} = \{G, W, \Lambda\}$ consists of 3 components: (i) a graph $G = \{V, E\}$, where edges $E_{ij} = 1$ if node X_i is a cause event or a parent of node X_j, $\{X_i, X_j\} \in V$, (ii) a set of window function W, where each window $w_{ij} \in W$ indicates the length of the recent history $[t - w_{ij}, t)$ that X_i would have an impact on X_j, and (iii) a set of intensity functions $\lambda_{i|\mathbf{u}} \in \Lambda$ for each node X_i, where u is the value of parent nodes of X_i.

Different from the vanilla PGEM, we also propose to capture the dependencies of the exogenous future events on the current consensus release. Let w_{r^-} and w_{r^+} define two windows to past and future revenue reports from consensus update (note that since we only consider Consensus here, we drop index c from w_r). Let t^- and t^+ be the times for the most recent past and future revenue reports at time t, respectively. Assuming consensus update rate $\lambda_i^C(t)$ at any given time t depends on whether a duration of w_{r^-} has passed since t^- and whether t^+ would be reached within w_{r^+} from t. Then, the intensity rate λ_i^C can be written as $\lambda_{i|\mathbf{u}}^C$, where $\mathbf{u}$ is the actual values of the causal factors for consensus updates. For example, if both past and future revenue report are the causal factors (or parent nodes in PGEM), then the intensity rate $\lambda^C_{i\tau|-\mp}$ signifies the rate at which event c occurs at any time τ given that event r has occurred at least once in the interval $[\tau - w_{r^-}, \tau)$ (hence $-$) and that r will not occur in $[\tau, \tau + w_{r^+})$ (hence $\mp$).

In a PGEM, graph G, window W, and intensity rates Λ can all be learned. We first talk about the learning of W and Λ, given G is known. Given one particular graph, we can optimize window W and intensity rates Λ by maximizing the following log likelihood function:

$$\begin{aligned}\log L(D_i^C) = &\sum_{\mathbf{u}} \left(-\lambda_{i|\mathbf{u}}^C D(\mathbf{u}) + N(c;\mathbf{u}) \ln\left(\lambda_{i|\mathbf{u}}^C\right)\right) \\ &+ \sum_{n=1}^{N} \log p(C_{in})\end{aligned} \tag{5}$$

where $N(c;\mathbf{u})$ is the number of times that c is observed in the dataset and that the condition $\mathbf{u}$ (from $2^{|\mathbf{U}|}$ possible parental combinations) is true in the relevant preceding or future windows, and $D(\mathbf{u})$ is the duration over the entire time period where the condition $\mathbf{u}$ is true. Formally,

$$N(c;\mathbf{u}) = \sum_{i=1}^{N} \mathbf{1}\left[l_i = c\right] \mathbf{1}_{\mathbf{u}}^{w_r}(t_i)$$

and

$$D(\mathbf{u}) = \sum_{i=1}^{N+1} \int_{t_{i-1}}^{t_i} \mathbf{1}_{\mathbf{u}}^{w_r}(t) dt,$$

where $\mathbf{1}[\cdot]$ is an indicator function, which takes value 1 if the condition is true and 0 otherwise. $\mathbf{1}_{\mathbf{u}}^{w_r}(t)$ is an indicator for whether $\mathbf{u}$ is true at time t as a function of the relevant windows w_r. From Equation 5, it is easy to see that the maximum likelihood estimates (MLEs) of the conditional intensity rates are $\hat{\lambda}_{c|\mathbf{u}} = N(c;\mathbf{u})/D(\mathbf{u})$.

Window learning can be found exactly if C only has one parent in PGEM. For a node x with a single parent Z, the log likelihood maximizing window w_{zx} either belongs to or is a left limit of a window in the candidate set $W^* = \{\hat{t}_{zx}\} \cup \max\{\hat{t}_{zz}\}$, where $\{\hat{t}\}$ denotes inter-event times in the datasets. It can be seen that the counts changes at the inter-events $\hat{t}_{zz}$, and they are step functions and therefore discontinuous at the jump points; this is the reason why the optimal window can be a left limit of an element in W^*. Hence, one can search for the best window, which maximizes the log-likelihood, in the set W^*. However, if c has more than one parents, the windows can be outside W^*. One heuristic is to search parents' window values one at a time, conditioned on previous windows.

Graph structure learning can be done with a forward and backward search (FBS) procedure to compute the max Bayesian information criterion [5], defined for a PGEM as:

$$\mathrm{BIC}(D_c^i) = \log \mathrm{L}(D_c^i) - \ln(T) 2^{|\mathbf{U}|}. \tag{6}$$

where T is the total time horizon in the datasets and $|\mathbf{U}|$ is the size of c's parent set. Given the BIC score, FBS conducts a forward search first and initializes the parental set of c to be empty, and then iteratively add one candidates parent nod to see if the resulting parental set increases the BIC score with learned W and Λ. If it is better than the current best score, FBS keeps the new parental set and check the next candidate. It runs until all variables have been tested. Then in the backward search phase, FBS iteratively tests if each candidate variable in the current parental set can be removed, i.e., if the rest of parents give a better BIC score. If so, the candidate parent is removed. After checking all candidates, FBS returns the resulting parental set as the learned parents. For more details on learning and consistency guarantees, we refer the readers to the original paper [5].

4.3 Consensus (C_{in}) and Revenue (Z_{in}) Modeling

In this subsection, we model the distributions of consensus magnitude and revenue magnitude based on their previous event magnitudes. We assume that the magnitudes do not depend on the event intensities at the time when the events happen; also, unlike the event intensity model, we do not consider the inter-dependencies of magnitudes among different companies, although our model can be easily extended to consider such interference and other external information.

Under the liquid market assumption, the consensus magnitude should reflect the market expectation on the revenue of the company. This means the last consensus magnitude should be the most accurate market estimate for the revenue, as long as there is at least one consensus update after the last revenue release. Therefore, given the last consensus value $C_i\left(t_{in}^Z\right)$, we assume that the revenue Z_{in} follows a normal distribution centered at $C_i\left(t_{in}^Z\right)$ with variance $(\sigma_i^Z)^2$, i.e.

$$Z_{in}|\mathcal{H}(t_{in}) \sim \mathcal{N}\left(C_i\left(t_{in}^Z\right), (\sigma_i^Z)^2\right);$$

and on the other hand, if there is no consensus change after the last revenue release, we simply use the last revenue magnitude as the mean, and the distribution is given by

$$Z_{in}|\mathcal{H}(t_{in}) \sim \mathcal{N}\left(Z_i\left(t_{in}^Z\right), (\sigma_i^Z)^2\right).$$

In a similar but slightly more complicated way, we model the distribution of consensus magnitude as follows

$$C_{in}|\mathcal{H}(t_{in}) \sim \mathcal{N}\left(\mu_{in}^C, (\sigma_i^C)^2\right), \tag{7}$$

where

$$\mu_{in}^C = C_{i,n-1} + \alpha\left(Z_i\left(t_{in}^C\right) - C_{i,n-1}\right)\mathbf{1}[\tau_i^C(t_{in}) < \tau_i^Z(t_{in})]. \tag{8}$$

Here, $Z_i\left(t_{in}^C\right)$ denotes the last revenue value announced before t_{in}^C and $\tau_i^C(t)$ and $\tau_i^Z(t)$ denote the time of the last consensus/revenue update before t, respectively. Intuitively, if there is no revenue update after the last consensus update, we assume that the value of this consensus update will be centered around the last value; otherwise, this consensus update will be also affected by the revenue update that happens after the last consensus update. In our experiments, we see that such simple adjustment is very effective in capturing the consensus changes.

4.4 Stock Return (R_{it}) Modeling

In practice, stock return is usually modeled as normal distribution with constant mean and standard deviation for simplicity, i.e., $R_{it} \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{p}\Sigma)$, where the mean and the variance-covariance matrix are constant [6, 10, 12]. This simple model cannot capture the impact of dynamic events which often leads to negative skewed returns. Therefore, we propose two models to incorporate effects of revenue release and consensus events to return modeling.

4.4.1 Normal with Adjusted Mean upon Events. To account for impact of event occurrences, we first assume the return expectation is affected by the most recent events as follows

$$\begin{aligned}\boldsymbol{\mu}_i = &\sum_{j,l}\alpha_{ijl}^Z\left(Z_j(t) - C_j(\tau_j^-)\right)\mathbf{1}[t = t_j^- + l]\\ &+\sum_j \alpha_{ij}^C(C_{jn} - C_{j(n-1)})\mathbf{1}[t = t_{jn}^C]\end{aligned} \tag{9}$$

where α_{ijl}^Z and α_{ij}^C are the coefficient of the impact of company j's release and consensus update on company i, l denotes the the number of days ahead of t, t_{jn}^C is the time of consensus update n.

4.4.2 Normal with Jump upon Events. The return time series models we considered so far does not take into account the fact that the stock prices become much more volatile upon the event occurrences. To account for this fact, we draw inspiration from the stochastic jump-diffusion process [16] for stock price modeling. If the log-stock price followed the stochastic process

$$\begin{aligned}d\log S_i(t) = \mu_i^R\,dt + \sigma_i\,dW_i^R(t) + \sum_{j=1}^N k_{i|j}^C(t)\,dq_j^C(t)\\ + \sum_{j=1}^N k_{i|j}^Z(t)\,dq_j^Z(t),\end{aligned}$$

where $q_i^C(t)$ and $q_i^Z(t)$ denote the counting process of consensus updates and earning releases of company i, respectively, and the jump magnitude $k_{i|j}^C(t)$ and $k_{i|j}^Z(t)$ followed a normal distribution that scales with the revenue surprise $P_j(t)$ or the consensus change $\Delta_j(t)$, i.e.

$$k_{i|j}^Z(t) \sim \mathcal{N}(\alpha_{i|j}^Z P_j(t), \beta_{i|j}^Z P_j^2(t)),$$

and

$$k_{i|j}^C(t) \sim \mathcal{N}(\alpha_{i|j}^C \Delta_j(t), \beta_{i|j}^C \Delta_j^2(t)),$$

then we would obtain the marginal distribution for $R_i(t)$ as a Gaussian distribution with mean

$$\mu_i(t) = \mu_i^R + \sum_{j=1}^N \mathbf{1}\{t = \tau_j^C(t)\}\alpha_{i|j}^C\Delta_j(t) + \sum_{j=1}^N \mathbf{1}\{t = \tau_j^Z(t)\}\alpha_{i|j}^Z P_j(t)$$

and variance

$$v_i(t) = \sigma_i^2 + \sum_{j=1}^N \mathbf{1}\{t = \tau_j^C(t)\}\beta_{i|j}^C\Delta_j^2(t) + \sum_{j=1}^N \mathbf{1}\{t = \tau_j^Z(t)\}\beta_{i|j}^Z P_j^2(t).$$

Based on the marginal distributions of $R_i(t)$ inspired by the jump-diffusion process, we now propose the joint distribution of $R(t)$ under the event models. We assume the correlation structure Θ among different companies does not change with time, and thus the joint distribution of $R(t)$ under the event model is given by

$$\boldsymbol{R}(t) \sim \mathcal{N}(\boldsymbol{\mu}(t), \boldsymbol{D}_{\boldsymbol{v}(t)}^{1/2}\boldsymbol{\Theta}\boldsymbol{D}_{\boldsymbol{v}(t)}^{1/2}),$$

where $D_{v(t)}$ denotes the diagonal matrix of $v(t)$.

Furthermore, we take into account the financial factor model: following [9], we perform PCA to filter the common factors and take the residual as the input of our learning framework. We then perform a three-stage algorithm to learn α's, β's and Θ one after another. The α's and β's are learned by least squares with ℓ_1 penalty, and Θ is learned by the thresholding operator following [9]. Note that since the mean model and the variance model are learned separately in a sequential order, each model can be replaced by a more complicated/advanced model with event adjustments. For example, we can replace the constant variance σ_i^2 in the variance expression $v_i(t)$ with an ARCH or GARCH [4, 7] model, and learn the ARCH/GARCH part and the event adjustment β's in a two-stage fashion.

5 EXPERIMENTS

In this section, we evaluate our proposed probabilistic framework using real market datasets and present experiment results to answer following questions:

(1) How good are the probabilistic models in recovering the dependency and probability of event occurrence?
(2) How effective are the event magnitude models capturing expected event magnitude?
(3) How profit if investment decisions are made based on inferred stock return accounting for event impact?

5.1 Experimental Setup

Our data collected from Refinitiv Eikon[1] includes 508 publicly traded companies in the US stock market and 220 companies in the Japan stock market. These companies are in the large capital group that consensus updates are relatively frequent due to investor interests. For each company, the data contains historical

[1]https://www.refinitiv.com/en

quarterly revenue release date and value, consensus adjustment date and value, and daily closing market price. The data ranges from 2007-01-02 to 2019-08-31. We use the data before 2017-01-01 as the training set to estimate model parameters and the rest to evaluate model performance. Upon parameter estimation, we infer future occurrence time of consensus updates and future values of revenue, consensus and return via 2000 runs of simulation. For good statistical properties, we used quarterly revenue growth over the previous quarter and the corresponding consensus in related calculation. The experiments are performed on a Macbook Pro with 2.6 GHz 6-Core Intel Core i7 and 32 GB 2400 MHz DDR4.

Note most existing deep/machine learning work on financial time series prediction is point estimation [14, 23]. Classical multivariate time series models in statistics/econometrics literature, e.g., vector autoregressive (VAR), do not capture event impacts to future values and usually assume normal distribution with constant mean and standard deviation. To the best of our knowledge, there is no directly comparable prior work to benchmark performance of this pioneering work quantitatively.

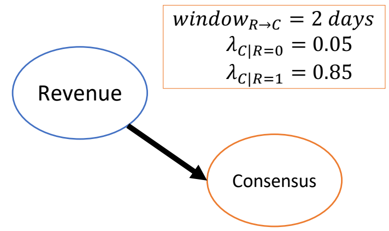
5.2 Experimental Results

Event Intensity. We first report the mean and standard deviation of per sample log likelihood values for consensus change intensity estimation in Table 1. We can see that our proposed PGEM obtains larger average likelihood values than the two extended MHP models in train and test for both US and Japan markets. This implies that data-driven method in learning temporal event dependency performs better than multivariate point process with domain knowledge. In addition, the likelihood values of US market is larger than that of Japan market which is due to richer consensus updates on US companies.

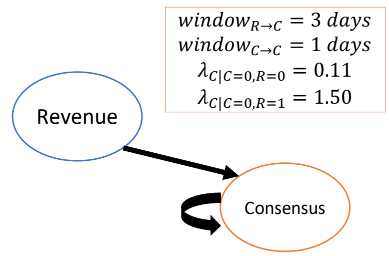
Table 1: Per sample log likelihood (mean ± std) for estimating intensity of consensus change

Market	Models	Train	Test
US	PGEM	-2.58±0.30	-2.45±0.37
	MHP (exp)	-3.32±0.41	-3.30±0.51
	MHP (sig)	-3.22±0.37	-3.26±0.51
Japan	PGEM	-2.83±0.71	-2.95±0.80
	MHP (exp)	-4.10±0.58	-4.09±0.70
	MHP (sig)	-4.15±0.66	-4.16±0.76

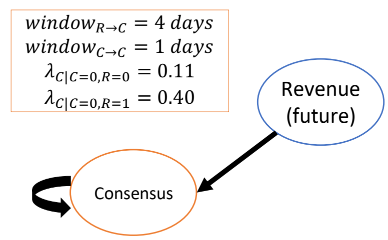
Next we visualize the learned graph structure for consensus change to illustrate event dependency. In Fig. 1, we show 3 most common learned structures along with the averaged parameters over all different companies. Fig. 1 (a) indicates that when revenue release occurred in the past 2 days, a consensus change is more likely to occur, where the consensus change intensity ($\lambda_{C|R=1}$ = 0.85) is about 17 times of that ($\lambda_{C|R=0}$ = 0.05) if there are no revenue release occurred in the same window. Fig. 1 (b) show the similar influence of revenue release to consensus change (1.50 vs 0.11) as well as consensus change following its previous update. Fig. 1(c) show that future revenue release also impact the rate of consensus update. Finally, the learned window size implies that the event



(a) #1 frequent graphical event structure



(b) #2 frequent graphical event structure



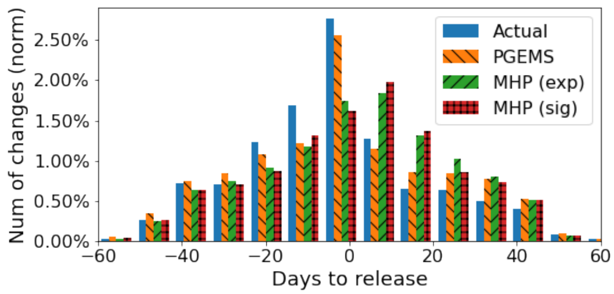
(c) #3 frequent graphical event structure

Figure 1: Top 3 most frequent learned PGEM models over Revenue and Consensus release events: (a), (b), and (c) account for 46.4%, 28.0%, and 16.8% of all the datasets, respectively.

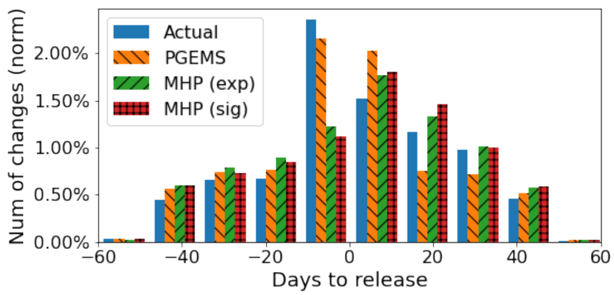
causal impact concentrates in a short time window ranging from 1 to 4 days.

Finally, we compare predicted and actual consensus update intensity over *dates to release*. For each company, we compute the average number of consensus changes over the 2000 simulations and summarize the value for all the companies as histogram normalized by the total events over *dates to release*. As shown in Fig. 2, consensus change highly concentrates around release dates and its intensity decreases quickly as *dates to release* increase. In addition, the learned consensus change densities of all our 3 models match that of actual adjustments, which validates that our proposed models can recover the probability of consensus adjustments. Moreover, PGEM performs better than MHPs, especially right before release time, based on the big spikes. This is consistent with the observation from log likelihood values in Table 1.

Event Magnitude. To illustrate the effectiveness of the consensus magnitude model, we compute the correlation of the actual consensus values and the predicted ones given the actual history

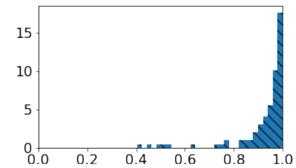


(a) US Market

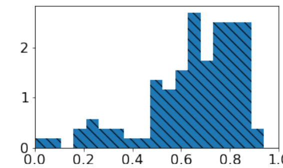


(b) Japan Market

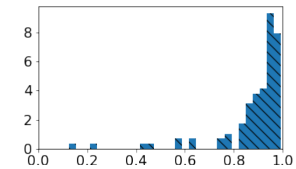
Figure 2: Normalized histogram of average number of consensus adjustments vs days to release.



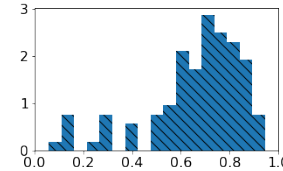
(a) US: learned α in Eq. (8)



(b) JP: learned α in Eq. (8)

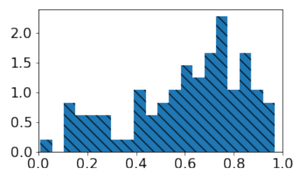


(c) US: fixed $\alpha = 0$ in Eq. (8)

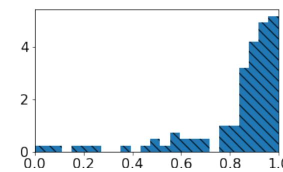


(d) JP: fixed $\alpha = 0$ in Eq. (8)

Figure 3: Consensus magnitude: histogram plot for correlation of predicted consensus magnitude and actual value.



(a) US



(b) JP

Figure 4: Revenue magnitude: histogram plot for correlation of predicted revenue magnitude and actual values.

up to the corresponding updates for each company and plot the (normalized) histograms of these correlations in Fig. 3. To validate the benefit of the adjustment term with the learnable coefficient α in the consensus magnitude model Eq. (8), we compare the distribution of correlations for the learned α (as shown in Fig. 3a and 3b) against that for the fixed $\alpha = 0$ (as shown in Fig. 3c and 3d). Note that $\alpha = 0$ means not using the adjustment term. We see that compared to the distribution without the adjustment term, the distribution with the adjustment tends to have higher correlation with the actual values. In particular for the US market, the correlation value is more concentrated around and closer to 1. This justifies the effectiveness of our model for consensus magnitude Eq. (8) with the adjustment term. In addition, the correlations in Fig. 3a are much more concentrated than those in Fig. 3b. This is because consensus of the selected US companies is much more frequently updated than those companies in the Japan market. In other words, we have richer data to train the model. On the contrary, as demonstrated in Fig. 4, the revenue prediction of US companies is more spread than that of Japan companies, which implies that market information has been efficiently taken into account via consensus update for the select US stocks.

Profitability. Ultimately, we evaluate our proposed probabilistic framework and novel probabilistic models for event and return modeling via key performance indicators (KPIs) of simulated portfolios. We use the results from our best event intensity model (PGEM) and its combination with variants of return distribution models. We create 2000 trials of daily returns in the test period for each companies and construct fully invested long-only portfolios that achieve a given objective, i.e., minimum volatility (Min Vol) or maximum Sharpe ratio (Max SR). The former one reflects the robustness of dynamic variance-covariance prediction and the latter one brings return forecasting into the picture. Based on the projected portfolio returns, we calculate 4 KPIs, i.e., annualized return (Annu. Ret), volatility (Vol), Sharpe ratio (SR), and maximum drawdown (MDD), as shown in Table 2. Here we select two common practices as baselines: 1) *Equal Weights* that distributes funding equally into all the stocks, which represents the market; 2) r_3 that portfolio returns are assumed as multivariate normal distribution with constant parameters.

For respective portfolio optimization objective, our proposed model r_1 performs the best which validates the benefits of taking into account dynamic event impacts in modeling both the mean and variance-covariance matrix of the return distribution. If we keep the variance-covariance matrix as constant as model r_2, compared to r_3, we observe that Annu. Ret increases for Max SR but decreases for Min Vol. This implies that return prediction is improved by incorporating event impact but the accuracy of variance-covariance estimation is scarified. The baseline *Equal Weights* portfolio that neither impact of crucial events nor historical information is leveraged to predict future stock returns performs the worst considering all the 4 KPIs. We observe similar KPIs from simulated portfolio

Table 2: Portfolio performance – US Market

Normal Dist.	Objective	Annu. Ret.	Vol	SR	MDD
r_1: $\mu_i(t)$,$v_i(t)$	Max SR	**9.70%**	15.87%	0.66	25.39%
	Min Vol	**9.07%**	**12.41%**	**0.76**	**18.63%**
r_2: $\mu_i(t)$	Max SR	7.54%	14.74%	0.57	21.76%
	Min Vol	7.89%	14.36%	0.60	23.04%
r_3: μ_0, $p\sigma$	Max SR	7.46%	14.07%	0.58	24.23%
	Min Vol	8.18%	12.19%	0.71	18.59%
Equal Weights		5.70%	14.68%	0.45	24.81%

using Japan market data and skip the presentation due to space limitation.

6 DISCUSSION AND FUTURE WORK

In this paper, we for the first time present a probabilistic framework to model the impact of stochastic events on multivariate financial time series. We apply the framework to model two types of events, i.e., quarterly revenue release and consensus change, and their impact to predict future stock returns and evaluate its performance using two sets of market data. Experiment results show that our model can recover the probability of event intensity and provide insight on casual event relationships as well as achieve improved portfolio performance over baseline models that are widely used in practice and literature for simplicity.

Although we demonstrate our proposed framework with these three financial variables, our probabilistic framework is generically applicable to other events, such as change of credit ranking, new product release, merging and acquisition, to name a few, as wells other financial variables, such as exchange rate, bond yield, and commodity future, etc. The events can be extracted from regular reports (e.g, quarterly earning reports), time series (e.g., based on thresholding the stock price series [19]), and news. Our model also can be easily extended to consider the inter-dependencies of magnitudes among different companies to reflect chain reaction of stock valuation [2] and other information. Moreover, all the probabilistic models in our generic probabilistic framework can be replaced by complicated/advanced model, such as neural networks for mean or variance-covariance prediction of stock returns, ARCH/GARCH models to predict return variance, or non-symmetric distributions for event and/or time series. However, financial forecasting problems usually involve large number of time series entities, limited sample sizes, correlated samples and low signal strengths which may lead to overfitting by using high complex models. In the future, we plan to deepen our models to applications related to high-frequency financial time series and interpretation of events from real-time noisy market news.

7 ACKNOWLEDGMENTS

This work was supported by the MIT-IBM Watson AI Lab, Refinitiv, an LSEG (London Stock Exchange Group) business, and Wells Fargo & Company through the Membership Program of the MIT-IBM Watson AI Lab. The views and conclusions expressed herein are those of the authors and should not be interpreted as that of IBM Research, Refinitiv or Wells Fargo & Company.

REFERENCES

[1] O. O. Aalen, O. Borgan, and H. K. Gjessing. 2008. *Survival and Event History Analysis: A Process Point of View.* Springer Science & Business Media, New York, NY, USA.

[2] Y. Aït-Sahalia, J. Cacho-Diaz, and R. J. Laeven. 2015. Modeling financial contagion using mutually exciting jump processes. *Journal of Financial Economics* 117, 3 (2015), 585–606.

[3] E. Bacry, I. Mastromatteo, and J.-F. Muzy. 2015. Hawkes processes in finance. *Market Microstructure and Liquidity* 1, 01 (2015), 1550005.

[4] L. Bauwens, S. Laurent, and J. VK Rombouts. 2006. Multivariate GARCH models: a survey. *Journal of applied econometrics* 21, 1 (2006), 79–109.

[5] D. Bhattacharjya, D. Subramanian, and T. Gao. 2018. Proximal graphical event models. In *Advances in Neural Information Processing Systems*. Montreal, Canada, 8136–8145.

[6] F. Black and M. Scholes. 2019. The pricing of options and corporate liabilities. In *World Scientific Reference on Contingent Claims Analysis in Corporate Finance: Volume 1: Foundations of CCA and Equity Valuation*. World Scientific, 3–21.

[7] T. Bollerslev, R. F. Engle, and D. B. Nelson. 1994. ARCH models. *Handbook of econometrics* 4 (1994), 2959–3038.

[8] D. R. Cox and P. A. W. Lewis. 1972. Multivariate point processes. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 3: Probability Theory*. 401–448.

[9] J. Fan, Y. Fan, and J. Lv. 2008. High dimensional covariance matrix estimation using a factor model. *Journal of Econometrics* 147, 1 (2008), 186–197.

[10] R. C. Grinold and R. N. Kahn. 2000. *Active portfolio management.* McGraw Hill New York.

[11] A. G. Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58, 1 (1971), 83–90.

[12] J. C. Hull. 2003. *Options futures and other derivatives.* Pearson Education India.

[13] Kazi T Islam, Christian R Shelton, Juan I Casse, and Randall Wetzel. 2017. Marked point process for severity of illness assessment. In *Machine Learning for Healthcare Conference*. 255–270.

[14] W. Jiang. 2020. Applications of deep learning in stock market prediction: recent progress. arXiv:2003.01859 [q-fin.ST]

[15] J. Liu, Y. Chen, K. Liu, and J. Zhao. 2017. Attention-Based Event Relevance Model for Stock Price Movement Prediction. In *Knowledge Graph and Semantic Computing. Language, Knowledge, and Intelligence*. 37–49.

[16] R. C. Merton. 1976. Option pricing when underlying stock returns are discontinuous. *Journal of financial economics* 3, 1-2 (1976), 125–144.

[17] M. Nickel and M. Le. 2020. Learning Multivariate Hawkes Processes at Scale. *arXiv preprint arXiv:2002.12501* (2020).

[18] J.-L. Prigent. 2001. Option pricing with a general marked point process. *Mathematics of Operations Research* 26, 1 (2001), 50–66.

[19] S. Xiao, J. Yan, M. Farajtabar, Le Song, X. Yang, and H. Zha. 2017. Joint modeling of event sequence and time series with attentional twin recurrent neural networks. *arXiv preprint arXiv:1703.08524* (2017).

[20] Shuai Xiao, Junchi Yan, Mehrdad Farajtabar, Le Song, Xiaokang Yang, and Hongyuan Zha. 2019. Learning time series associated event sequences with recurrent point process networks. *IEEE transactions on neural networks and learning systems* (2019).

[21] S. Xiao, J. Yan, X. Yang, H. Zha, and S. M Chu. 2017. Modeling the intensity function of point process via recurrent neural networks. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[22] H. Xu, D. Luo, and H. Zha. 2017. Learning Hawkes processes from short doubly-censored event sequences. In *Proceedings of the International Conference on Machine Learning (ICML)*. 3831–3840.

[23] D. Zhou, L. Zheng, Y. Zhu, J. Li, and J. He. 2020. Domain Adaptive Multi-Modality Neural Attention Network for Financial Forecasting. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) *(WWW '20)*. Association for Computing Machinery, New York, NY, USA, 2230–2240.

[24] K. Zhou, H. Zha, and L. Song. 2013. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *Artificial Intelligence and Statistics*. 641–649.

[25] K. Zhou, H. Zha, and L. Song. 2013. Learning triggering kernels for multi-dimensional Hawkes processes. In *Proceedings of the International Conference on Machine Learning (ICML)*. 1301–1309.