ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9.
Available from http://www.i6doc.com/en/.

Revisiting the Mark Conditional Independence Assumption in Neural Marked Temporal Point Processes

Tanguy Bosser and Souhaib Ben Taieb

University of Mons - Department of Computer Science
Avenue Victor Maistriau, 15, Mons - Belgium

Abstract. Learning marked temporal point process (TPP) models involves modeling both the event arrival times as well as their associated labels, referred to as marks. The recent introduction of deep learning techniques to the field led to better modeling of event sequences thanks to more flexible neural TPP models. However, some of these models make the assumption that event marks are independent of event times given the history of the process, which may not be valid in many applications. We relax this assumption and explicitly parametrize the mark distribution as a function of the current event time. We show that our approach achieves improved performance in predicting future marks compared to baselines on multiple real-world event sequence datasets, without affecting the performance on event time prediction.

1 Introduction

A broad range of systems are often characterized by sequences of discrete events taking place at irregular time intervals. Common examples may include users activity on a social media platform, e-commerce transactions, or earthquakes manifestations. Given past realizations of a system of interest, one may be interested in capturing the correlations among past event occurrences to enable prediction of future ones. In practice, these events are often associated to additional information, such as discrete classes, or *marks*, that we may wish to infer along the corresponding timestamp. Temporal Point Processes (TPP) [1] provide a powerful mathematical framework for modeling these streams of asynchronous and cross-correlated event data. However, classical parametrizations of TPP models, such as the Hawkes process [2], have often been criticized for their lack of flexibility in modeling complex event dynamics [3]. To increase the models' capacity, deep learning methods have been introduced to the field of TPP, including RNN [4] and self-attention mechanisms [5, 6]. Among these neural TPP architectures, LogNormMix [7] has proven itself to be a strong baseline in fitting the distribution of future event arrival times, often outperforming more recent architectures [8]. However, by assuming that the marks are conditionally independent of time given a process history, LogNormMix can hinder performance in capturing the dynamic of mark occurrences if this assumption is not valid. In this work, we provide a simple yet useful modification in the parametrization of LogNormMix to account for the dependence of future marks on time. Through

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

experiments on 6 real-world datasets, we show that our approach often outperforms LogNormMix in predicting future marks, while keeping similar fitting capabilities when estimating the distribution of future arrival times.

2 Background and notations

Marked temporal point processes (MTPP) are stochastic processes whose realizations consist in sequences of n discrete events $\mathcal{S} = \{e_i = (t_i, k_i)\}_{i=1}^n$ observed within a fixed window $[0, T]$. For each event e_i, t_i corresponds to the event *arrival time* with $0 \leq t_1 < ... < t_n \leq T$, while $k_i \in \mathcal{K} = \{1, ..., K\}$ is the associated *mark*, or class to which the event belongs. Note that $\mathcal{S}$ can be equivalently represented as $\{e_i = (\tau_i, k_i)\}_{i=1}^n$, where $\tau_i = t_i - t_{i-1}$ is the event *inter-arrival time*. We will use both representations interchangeably throughout the paper. In an MTPP, the occurrence of future arrival times and marks can be fully characterized through the conditional joint distribution $f(t, k|\mathcal{H}_t)$, where $\mathcal{H}_t = \{(t_i, k_i \in \mathcal{S})|t_i < t\}$ is the process history up to time t. For clarity, we will employ the notation '$*$' of [1] to indicate dependence on $\mathcal{H}_t$, i.e. $f(t, k|\mathcal{H}_t) = f^*(t, k)$. Provided a parametric form of $f^*(t, k; \boldsymbol{\theta})$, the most common approach to learning the set of parameters $\boldsymbol{\theta}$ is achieved by negative log-likelihood (NLL) minimization. Given a sequence $\mathcal{S}$ of n events, and by noting $f^*(t, k; \boldsymbol{\theta}) = f^*(t; \boldsymbol{\theta})p^*(k|t; \boldsymbol{\theta})$ with $p^*(k|t; \boldsymbol{\theta})$ being the conditional distribution of marks, the NLL objective writes [9]:

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{S}) = \underbrace{-\sum_{i=1}^{n} \log f^*(t_i; \boldsymbol{\theta}) + \Lambda^*(T; \boldsymbol{\theta})}_{\text{NLL-T}} \underbrace{-\sum_{i=1}^{n} \log p^*(k_i|t_i; \boldsymbol{\theta})}_{\text{NLL-M}}, \tag{1}$$

where $\Lambda^*(T; \boldsymbol{\theta}) = \sum_{k=1}^{K} \int_{t_n}^{T} \frac{f^*(t,k;\boldsymbol{\theta})}{1-F^*(t;\boldsymbol{\theta})} dt$ accounts for the fact that no event was observed in $]t_n, T]$, with $F^*(t; \boldsymbol{\theta})$ being the conditional cumulative distribution of arrival times. From (1), we can see that learning a MTPP from a sequence essentially reduces to two learning problems: 1) learning the conditional distribution of arrival-times $f^*(t)$ through the NLL-T term, and 2) learning the conditional distribution of marks $p^*(k|t)$ through the NLL-M term. In the following, we refer to these two problems as the *time prediction* and *mark prediction* tasks, respectively.

3 Our TPP model

Modeling future inter-arrival times. Suppose t_{i-1} is the last observed event in a sequence $\mathcal{S}$. To model the density of the next inter-arrival time $\tau = t - t_{i-1}$, we employ the LogNormMix (LNM) model [7], which defines $f^*(\tau)$ as a mixture of log-normal distributions. Specifically, given $\boldsymbol{h} \in \mathbb{R}^{d_h}$, a representation of the

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

history $\mathcal{H}_t$ of a query event $e = (t, k)$ with $t > t_{i-1}$, LNM defines $f^*(\tau)$ as:

$$f^*(\tau) = \sum_{m=1}^{M} p_m \frac{1}{\tau \sigma_m \sqrt{2\pi}} \exp\left(\frac{(\log \tau - \mu_m)^2}{2\sigma_m^2}\right), \tag{2}$$

where $p_m = \text{Softmax}\left(\mathbf{W}_m \boldsymbol{h} + \boldsymbol{b}\right)_m$ corresponds to the probability that τ was generated by the m^{th} mixture component, while $\mu_m = \boldsymbol{w}_\mu \boldsymbol{h} + b_\mu$ and $\sigma_\mu = \exp(\boldsymbol{w}_\sigma \boldsymbol{h} + b_\sigma)$ are the mean and standard deviation of the m^{th} mixture component, respectively. $\mathbf{W}_m \in \mathbb{R}^{M \times d_h}$, with M being the number of mixture components, $\boldsymbol{b} \in \mathbb{R}^M$, $\boldsymbol{w}_\mu, \boldsymbol{w}_\sigma \in \mathbb{R}^{d_h}$, and b_μ, b_σ are scalars. Similar to [4, 7], the history embedding $\boldsymbol{h}$ is obtained sequentially by applying a GRU on $\mathcal{H}_t$, i.e. $\boldsymbol{h} = \text{GRU}(\boldsymbol{e}_{i-1}, \boldsymbol{h}_{i-2})$, where $\boldsymbol{e}_{i-1} = [\boldsymbol{e}_{i-1}^t || \boldsymbol{e}_{i-1}^k] \in \mathbb{R}^{d_e}$ is an encoding of $e_{i-1} = (t_{i-1}, k_{i-1})$. In this last expression, $\boldsymbol{e}_{i-1}^t \in \mathbb{R}^{d_t}$ and $\boldsymbol{e}_{i-1}^k \in \mathbb{R}^{d_k}$ are encodings of t_{i-1} and k_{i-1} respectively, while $||$ refers to the concatenation operator. Following [5, 6, 8], we construct $\boldsymbol{e}_{i-1}^t$ using a vector of sinusoidal functions on t_{i-1}, and $\boldsymbol{e}_{i-1}^k$ by passing k_{i-1} through a mark embedding layer, i.e. $\boldsymbol{e}_{i-1}^k = \mathbf{W}_K \boldsymbol{k}_{i-1}$, where $\mathbf{W}_K \in \mathbb{R}^{d_k \times K}$ and $\boldsymbol{k}_{i-1} \in \mathbb{R}^K$ is the one-hot encoding of k_{i-1}. Finally, $\boldsymbol{h}_{i-2}$ corresponds to the history embedding of e_{i-1}. Despite its apparent simplicity, modeling the conditional distribution of inter-arrival times with a mixture of log-normal distributions has often proven to be a strong inductive bias in real-world event sequence datasets [6, 8, 10].

Modeling future marks. LNM defines a categorical distribution over future marks using a softmax transformation applied to a feed-forward layer on the history embedding,

$$p^*(k|t) = p^*(k) = \text{Softmax}\left(\mathbf{W}_2 \text{tanh}\left(\mathbf{W}_1 \boldsymbol{h} + \boldsymbol{b}_1\right) + \boldsymbol{b}_2\right), \tag{3}$$

where $\mathbf{W}_2 \in \mathbf{R}^{K \times d_1}$, $\mathbf{W_1} \in \mathbb{R}^{d_1 \times d_h}$, $\boldsymbol{b}_1 \in \mathbb{R}^{d_1}$, and $\boldsymbol{b}_2 \in \mathbb{R}^K$ are learnable parameters. Note that in this expression, the mark distribution is made explicitly independent of the time given the history of the process. While this simple time-independent approach had also been used in earlier works [4], we believe that the knowledge of an event arrival time conveys useful information regarding the occurrence of its associated mark. In this regard, we aim to define the probability distribution over future marks as being explicitly dependent on the present (inter-arrival) time, as well as on the process' history. Given an encoding $\boldsymbol{e}^t \in \mathbb{R}^{d_t}$ of a query time $t \geq t_{i-1}$ and $\boldsymbol{h}$, we define the conditional distribution of marks as:

$$p^*(k|t) = \text{Softmax}\left(\mathbf{W}_2 \text{ReLU}\left(\mathbf{W}_1 [\boldsymbol{h} || \boldsymbol{e}^t] + \boldsymbol{b}_1\right) + \boldsymbol{b}_2\right), \tag{4}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_1 \times (d_h + d_t)}$. While we chose to encode $\boldsymbol{e}_{i-1}^t$ as a vector of sinusoidal functions to generate $\boldsymbol{h}$, we empirically found that setting $\boldsymbol{e}^t = \log \tau$ worked best in modeling $p^*(k|t)$. The definition in (4) is flexible and can capture non-monotonic dynamics of the mark distribution between two events. In summary, our approach that we call **CondLogNormMix (CLNM)**, models the joint distribution of inter-arrival times and marks $f^*(\tau, k) = f^*(\tau) p^*(k|\tau)$ with $f^*(\tau)$ given by (2), and $p^*(k|\tau)$ given by (4).

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

4 Experiments

Datasets. Our experiments are based on six marked datasets frequently encountered in the TPP literature: LastFM [11], MOOC [11], Github [12], Stack overflow [4], Retweets [3], and Reddit [11]. Each dataset is filtered to contain at most its 50 most represented marks, and sequences containing less than two events are discarded. Additionally, the event arrival times are scaled in the interval $[0, 10]$ to avoid numerical instabilities, and we only keep 50% of the sequences originally contained in Reddit and Retweets to reduce computational time.

Experimental setup. We compare our CLNM model on the *mark prediction task*, described in Section 2, against three baselines: 1) The original formulation of LNM, where $p^*(k|t)$ is defined by (3), 2) A multivariate Hawkes process [2], and 3) The flexible self-attention decoder (SA/MC) introduced in [8]. Both Hawkes and SA/MC model the dependency of marks on time through the intensity function. Conversely, CLNM is an intensity-free TPP model which parametrizes the inter-arrival times and marks distributions separately. For each dataset, we randomly split the sequences into 5 train/validation/test splits following a 60%/20%/20% partition, respectively. The models are trained to minimize the NLL in (1) on the training sequences using mini-batch gradient descent. Optimization is carried out using the Adam optimizer with a learning rate of 10^{-3}. For all models with the exception of Hawkes, we set $d_t = 4$, $d_k = 4$, and $d_h = 16$. Additionally, for LNM and CLNM, $d_1 = 16$, and the number of mixture components is fixed at $M = 16$.

Evaluation metrics. As this paper mostly relates to the distribution of future marks, we mainly focus our analysis on the mark prediction task. To this end, we report for each dataset the NLL-M as defined in (1) averaged over all test sequences, as well as the F1-score, where we predict the next mark using $\tilde{k} = \operatorname{argmax}_k p^*(k|t)$. Moreover, to evaluate the statistical consistency between the mark predictions and the actual observations, we asses the calibration of the different approaches with respect to the mark distribution by means of the Expected Calibration Error (ECE) and reliability diagrams [13]. Finally, for completeness, we also report the NLL-T term in (1) to compare the different baselines on the time prediction task. Lower NLL-M, NLL-T and ECE is better, while higher F1-score is better.

Results and discussion. Table 1 displays the baseline results with respect to the aforementioned metrics on all datasets, averaged over the different splits. We observe that CLNM often yields improvements on the mark prediction task in comparison to the time-independent LNM model, as illustrated by lower NLL-M and ECE, and a higher F1-score on most datasets. This suggests that the inclusion of an explicit time dependency in the mark distribution makes our approach more amenable to capturing complex dynamics of mark occurrences in real-world datasets compared to the original formulation of LNM. However,

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

	NLL-M					
	LastFM	MOOC	Github	Stack Overflow	Reddit	Retweets
LNM	690.4 (17.5)	88.32 (1.46)	115.94 (15.3)	106.41 (0.68)	44.52 (1.19)	83.47 (0.25)
CLNM (Ours)	670.37 (16.3)	**77.76 (0.87)**	**112.38 (14.96)**	**103.03 (0.67)**	**42.24 (1.18)**	**83.16 (0.22)**
Hawkes	**514.13 (16.19)**	112.14 (1.26)	122.44 (15.5)	114.99 (0.75)	59.07 (1.06)	90.46 (0.34)
SA/MC	844.95 (23.67)	95.06 (1.6)	131.8 (18.75)	105.31 (0.35)	48.3 (1.61)	84.0 (0.29)
	ECE					
	LastFM	MOOC	Github	Stack Overflow	Reddit	Retweets
LNM	0.17 (0.03)	0.08 (0.02)	0.14 (0.0)	0.08 (0.02)	**0.03 (0.0)**	0.07 (0.01)
CLNM (Ours)	0.07 (0.02)	**0.03 (0.0)**	0.15 (0.0)	**0.02 (0.0)**	**0.03 (0.0)**	**0.06 (0.0)**
Hawkes	**0.03 (0.0)**	0.14 (0.0)	**0.07 (0.02)**	0.09 (0.02)	0.05 (0.0)	0.12 (0.0)
SA/MC	0.46 (0.02)	0.12 (0.02)	0.18 (0.01)	**0.02 (0.0)**	0.06 (0.0)	0.09 (0.0)
	F1-score					
	LastFM	MOOC	Github	Stack Overflow	Reddit	Retweets
LNM	0.2 (0.01)	0.4 (0.0)	0.57 (0.01)	0.33 (0.0)	**0.81 (0.0)**	**0.6 (0.0)**
CLNM (Ours)	0.22 (0.01)	**0.51 (0.0)**	**0.6 (0.01)**	**0.35 (0.0)**	**0.81 (0.0)**	**0.6 (0.0)**
Hawkes	**0.3 (0.0)**	0.29 (0.0)	0.54 (0.01)	0.32 (0.0)	**0.81 (0.0)**	0.55 (0.0)
SA/MC	0.02 (0.01)	0.33 (0.0)	0.47 (0.02)	0.34 (0.0)	0.8 (0.0)	**0.6 (0.0)**
	NLL-T					
	LastFM	MOOC	Github	Stack Overflow	Reddit	Retweets
LNM	**-1330.37 (57.5)**	**-308.49 (3.63)**	**-380.67 (60.08)**	**-91.05 (1.42)**	**-96.31 (2.11)**	**-600.22 (3.22)**
CLNM (Ours)	-1323.98 (58.94)	-308.27 (3.77)	-379.21 (59.45)	-90.99 (1.37)	-96.27 (2.01)	-598.36 (3.7)
Hawkes	-1189.48 (55.2)	-235.9 (3.09)	-308.42 (57.77)	-83.18 (1.4)	-75.6 (2.06)	-553.18 (1.91)
SA/MC	-1026.48 (42.43)	-239.7 (2.76)	-320.67 (55.68)	-87.12 (1.44)	-87.32 (2.09)	-576.61 (1.8)

Table 1: Baselines results on different datasets. Standard error across all splits is reported in parenthesis. Best results are highlighted in bold, and further underlined if the difference with the second best is larger than a standard error.

despite being dominated by other baselines for most datasets on the mark prediction task, we notice that the multivariate Hawkes model achieves top performance on the LastFM dataset. Events in LastFM are usually highly clustered per mark, and we believe that the assumption of strictly additive influence of past events in a Hawkes process makes it more prone to capture such dynamics. Nonetheless, while improving performance on the mark prediction task, CLNM does not induce significant changes in NLL-T compared to LNM, which remains the strongest baseline on the time prediction task. On Figure 1, we show the reliability diagrams of the mark distribution of all baselines on the LastFM and MOOC datasets. Consistently with the ECE values of Table 1, CLNM shows improved calibration compared to LNM, indicated by the bins' accuracy better aligned with the diagonal. Specifically, we notice a higher number of samples falling into the high probability bins for CLNM, meaning that the latter assigns more confidence to correct predictions. For instance, on LastFM, LNM makes no correct prediction with probability in [0.8,1].

5 Conclusion

In this work, we discarded the simplifying assumption of LNM that models the distribution of future marks as being conditionally independent of the time given the history of the process. Through experiments on 6 real-world event sequence datasets, we show that our approach often outperforms the original time-independent LogNormMix model on the mark prediction task, along other time-dependent baselines. This suggests that, despite its simple formulation,

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

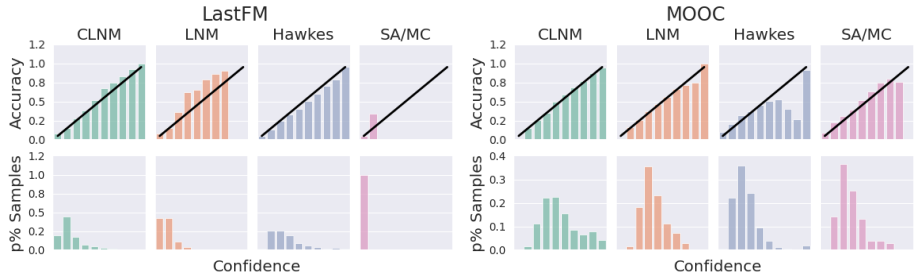

Fig. 1: Reliability diagrams of the distribution of marks on LastFM and MOOC. Front rows depict a model's average accuracy per bin, while bottom rows show the average proportion of samples falling per bin.

CLNM is more amenable to capture mark dependencies across events, without affecting the strong predictive power of LogNormMix on the time prediction task. We hope that our results will inspire new developments regarding the modeling of future marks in the field of TPPs.

References

[1] Daryl Daley and David Vere-Jones. An introduction to the theory of point processes volume ii: General theory and structure, 2007.

[2] Alan G Hawkes. Point spectra of some mutually exciting point processes, 1971. Journal of the Royal Statistical Society: Series B, 33(3).

[3] Hongyuan Mei and Jason Eisner. The neural hawkes process: A neurally self-modulating multivariate point process, 2016. NeurIPS.

[4] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector, 2016. SIGKDD.

[5] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process, 2020. ICML.

[6] Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive hawkes processes, 2020. ICML.

[7] Oleksandr Shchur, Marin BiloÅ¡, and Stephan GÃ¼nnemann. Intensity-free learning of temporal point processes, 2019. ICLR.

[8] Joseph Enguehard, Dan Busbridge, Adam Bozson, Claire Woodcock, and Nils Y. Hammerla. Neural temporal point processes for modelling electronic health records, 2020. ML4H.

[9] Jakob Gulddahl Rasmussen. Lecture notes: Temporal point processes and the conditional intensity function, 2018.

[10] Haitao Lin, Cheng Tan, Lirong Wu, Zhangyang Gao, and Stan. Z. Li. An empirical study: Extensive deep temporal point process, 2021.

[11] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks, 2019. SIGKDD.

[12] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Representation learning over dynamic graphs, 2018. ICLR.

[13] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks, 2017. ICML.