INTERPRETABLE NEURAL TEMPORAL POINT PROCESSES FOR MODELLING ELECTRONIC HEALTH RECORDS

Bingqing Liu
University of Chinese Academy of Science; Academy of Mathematics and System Science, CAS
liubingqing20@mails.ucas.ac.cn

ABSTRACT

Electronic Health Records (EHR) can be represented as temporal sequences that record the events (medical visits) from patients. Neural temporal point process (NTPP) has achieved great success in modeling event sequences that occur in continuous time space. However, due to the black-box nature of neural networks, existing NTPP models fall short in explaining the dependencies between different event types. In this paper, inspired by word2vec and Hawkes process, we propose an interpretable framework inf2vec for event sequence modelling, where the event influences are directly parameterized and can be learned end-to-end. In the experiment, we demonstrate the superiority of our model on event prediction as well as type-type influences learning.

1 INTRODUCTION

Event sequence is a ubiquitous data structure in real world, such as user behavior sequences, error logs, purchase transaction records and electronic health records (Mannila et al., 1997; Liu et al., 1998; Zhou et al., 2013; Choi et al., 2016; Liu & Huang, 2023). An event can be generally represented as a tuple, including the event type and occurrence time, e.g., (well-child visit, 2024/02/01). Inside the event sequence, various types of events often exhibit complex temporal patterns, making the type-type influences discovering even challenging. Temporal point process has been a popular and principled tool for event sequence modeling Shchur et al. (2021). Due to the high capacity of deep networks, neural temporal point process models have been intensively devised and have demonstrated superior performance for tasks such as event prediction (Du et al., 2016; Omi et al., 2019; Waghmare et al., 2022; Soen et al., 2021; Zhou & Yu, 2023; Mei & Eisner, 2017; Chen et al., 2018). However, their black-box nature makes most of them lack transparency and prevents them from explaining their decisions (Danilevsky et al., 2020; Minh et al., 2022; Linardatos et al., 2020).

Some attempts are made to make the event sequence model more explainable. AutoNPP (Zhou & Yu, 2023) adopts the additive form of the intensity function to capture the historical impacts, like Hawkes process. NRI-TPP (Zhang & Yan, 2021) leverages the variational inference to recover the underlying event dependencies by message passing graph and recurrent neural network (RNN). CAUSE (Zhang et al., 2020b) learns the Granger causality between event types by attribution methods, namely integrated gradients. Attention mechanism is also widely used (Zuo et al., 2020; Zhang et al., 2020a; Choi et al., 2016; Dash et al., 2022; Gu, 2021). However, these models either trade accuracy and efficiency for interpretability, or are only designed for discovering the token-wise influcences, or are model-specific, i.e., can not be applied to other models. Moreover, for the mostly widely used attention mechanism, there exists layer and head inconsistency about its interpretability, i.e., different attention layers and heads have different attention scores, which undermines its practical utility. To learn type-type influences, we draw inspiration from word2vec (Mikolov et al., 2013), which learns semantic vector representations that can help group similar words. Though this vanilla interpretability can not indicate the type-type influences, the mutual influences could be reflected if we take a further step: create a vector space for each event type. As a result, distributed representations of event types in the vector space of event type k can help group event types that have similar influences on k.

2 METHOD

In this section, we detail the proposed type-type influences learning framework, namely inf2vec. The event sequence model has three modules: the embedding layer, the sequence encoder, and the event decoder. Besides word2vec, our model also draws inspiration from Hawkes process (Hawkes, 1971a;b). Both of us model the event dynamics as an influence-driven process in the view of event types. To capture the impacts of historical events $\{(k_i, t_i)|t_i < t\}$, Hawkes process specifies the conditional intensity function as follows,

$$\lambda_k(t) = \mu_k + \sum_{t_i<t} \alpha_{k,k_i} \exp(-\beta_{k,k_i}(t - t_i)) \tag{1}$$

where $\mu_k \geq 0$ is the base intensity, $\alpha_{k,k_i} \geq 0$ is the coefficient indicating how significantly event k_i will influence the occurrence of event k, and $\beta_{k,k_i} \geq 0$ shows how the influence decays over time. The schematic representation of our model is illustrated in Fig.1.

2.1 EMBEDDING LAYER

Global vs local embedding. In conventional neural temporal point process models, the embedding layer assigns each event type a vector representation and the learned type representations can naturally group similar event types. However, it's hard for this kind of embedding to explain the dependencies between different event types. For example, which event type influences the given event type k most? One may turn to techniques like dot product to find the most influencing event type but again, it can only find the most similar rather than influencing event type. In fact, the conventional embedding can be considered as the global embedding, i.e., the vector representations of different types are in the same vector space. Our idea is to create K vector spaces and in each, an event type will have a vector representation, which we call local embedding. This kind of embedding can naturally reflect the relationships between different event types. An event type can have different vector representations in different vector spaces, indicating that it can have different influences on different event types. Moreover, given the vector space of event type k, close vector representations in this space indicate that the corresponding event types have similar impacts on event type k.

The local embedding is inspired by Hawkes process (Eq.1), where the impact of historical event (k_i, t_i) on the occurrence of event k is explicitly characterized by two learnable parameters α_{k,k_i} and β_{k,k_i}. From the perspective of embedding, the two parameters $[\alpha_{k,k_i}, \beta_{k,k_i}]$ can be considered as the vector representation of event type k_i in the vector space of event type k, with the embedding dimension being 2. More generally in this work, we set the embedding dimension as a hyperparameter d and use notation $\boldsymbol{z_m^k}(k_i) \in \mathbb{R}^d$ to denote the embedding of event type k_i in the vector space of event type k. And we use temporal embedding function $\boldsymbol{z_t^k}(t_i)$ to embed the timestamp t_i. Then in the context of event type k, we obtain the event embedding by concatenating the type and time embedding,

$$\boldsymbol{e^k}(i) = \boldsymbol{z_m^k}(k_i)||\boldsymbol{z_t^k}(t_i) \tag{2}$$

2.2 SEQUENCE ENCODER

Global vs local encoding. To capture the impacts of historical events $\{(k_j, t_j)\}_{j=1}^{i}$, existing sequence encoders encode them into one singe vector, which can be summarized as follows:

$$\boldsymbol{h_i} = Seq2Vec(\boldsymbol{e}(1) \cdots, \boldsymbol{e}(j), \cdots, \boldsymbol{e}(i)) \tag{3}$$

where $\boldsymbol{e}(j)$ is the embedding of event (k_j, t_j) and the backbone network $Seq2Vec$ is usually realized by RNN (Chung et al., 2014) or Transformer (Vaswani et al., 2017). The conventional encoder is devised specifically for the global event embedding, which we call global encoding. To handle the local event embedding, we here propose type-wise encoder:

$$\boldsymbol{h_i^k} = Seq2Vec^k(\boldsymbol{e^k}(1) \cdots, \boldsymbol{e^k}(j), \cdots, \boldsymbol{e^k}(i)) \tag{4}$$

which performs separate history encoding for different event types. The underlying rationale for the type-wise encoder is that each event type can concentrate on the historical events of its own interests, which we call local encoding. For example, if only the event (k_j, t_j) has impact on the occurrence of event k, then the history encoding $\boldsymbol{h_i^k}$ can only encode the information from $\boldsymbol{e^k}(j)$ and ignore

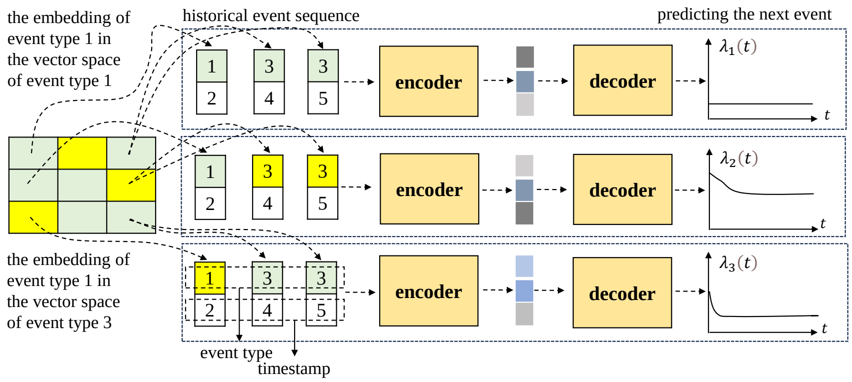

Figure 1: An overview of our proposed type-type influences learning framework. The framework creates separate embedding, encoding and decoding space for each event type. In the illustrated example, brighter color in the embedding layer means stronger influence and we see event of type 2 is more likely to occur.

the other uninterested events. As a comparison, the conventional encoder tries to summarize all historical information into one single vector $\boldsymbol{h_i}$. From this point of view, we can regard the local encoding $\{\boldsymbol{h_i^k}\}_{k=1}^K$ as the information decoupling (from the perspective of event types) of the global encoding $\boldsymbol{h_i}$, with each local encoding $\boldsymbol{h_i^k}$ only containing the historical information that the event type k is interested in.

2.3 Event Decoder

Global vs local decoding. With history representation obtained from the sequence encoder, we are about to decode the next event (the $(i+1)$-th event). In existing neural temporal point process models, the next event distribution is mostly characterized by the conditional intensity function, which we summarize as follows,

$$\lambda_k(t) = \sigma(NN_k(\boldsymbol{h_i}, t)) \tag{5}$$

where σ is an activation function to ensure the positive constraint of the intensity function and NN_k is a neural network, e.g., multi-layer perceptron. We see that in the conventional decoder, the global history encoding $\boldsymbol{h_i}$ is shared across different intensity decoders $\{NN_k\}_{k=1}^K$, which we call global decoding. Each intensity decoder takes out the information of interests from the global history encoding $\boldsymbol{h_i}$ and generate the corresponding intensity $\lambda_k(t)$. But in our framework, extracting information of interests is unnecessary as the global history encoding has already been decoupled into local history encoding in the view of event types in the encoding stage. Therefore, we can simply replace the global history encoding as the local history encoding, i.e.,

$$\lambda_k(t) = \sigma(NN_k(\boldsymbol{h_i^k}, t)) \tag{6}$$

we call it local decoding. In temporal point process, the next event distribution can be also described by cumulative hazard function (Omi et al., 2019), probability density function (Shchur et al., 2021), etc. They have quite different functional formulations comparing to Eq.5. But without loss of generality, we can accordingly adapt them to our framework.

3 Experiment

In this part, we design experiments to answer the following questions: **Q1**, What's the performance of inf2vec on standard prediction tasks? **Q2**, How's the quality of the learned type-type influences?

3.1 Datasets

Three publicly available EHR datasets are used, namely SynEHR1, SynEHR2, and MIMIC (Enguehard et al., 2020; Waghmare et al., 2022). Each dataset has a number of event sequences and

Table 1: Comparison of weighted F1 score and mean absolute error on event prediction. The results are averaged by 10 runs and we use bold numbers to indicate the best performance.

Model	Haw9		HawC9		SynEHR1		SynEHR2		MIMIC	
	F1	MAE	F1	MAE	F1	MAE	F1	MAE	F1	MAE
NHP	22.8	0.46	36.5	0.41	56.3	1.93	53.3	3.11	62.3	0.41
FullyNN	22.6	0.40	36.2	0.38	59.4	1.35	56.0	2.42	63.9	0.30
SAHP	22.5	0.44	36.3	0.39	58.9	1.85	55.7	2.81	63.3	0.33
THP	23.7	0.43	37.4	0.39	58.7	1.77	55.3	2.73	63.5	0.35
JTPP	23.1	0.39	38.5	**0.34**	59.6	**1.05**	56.1	2.11	64.9	0.27
Inf2vec	**24.5**	**0.37**	**39.1**	0.36	**60.2**	1.11	**56.4**	**2.01**	**65.3**	**0.24**

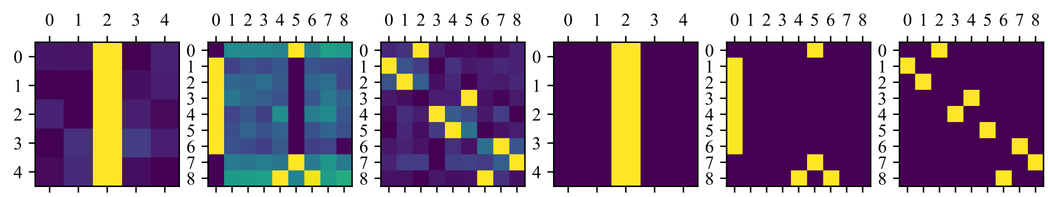

Figure 2: The illustration of our learned local embeddings (after dimension reduction) over dataset Haw5, Haw9 and HawC9 (the first three plots), coordinates (x, k) denotes the embedding of event type x in the context of event type k. The last three plots are the ground truth influences and coordinates (x, k) denotes the influence of x on k, brighter color means stronger influence.

each sequence records the medical events from a patient. They have 6, 178, and 75 event types, respectively. To evaluate the quality of the learned type-type influences, we additionally use three synthetic datasets, Haw5, Haw9 and HawC9, which are all simulated by Hawkes process (Eq.1) but have different parametric settings. Technically, we use the open-source python library tick [1] to simulate the Hawkes process. These three Hawkes datasets have 5, 9 and 9 event types, respectively. Each dataset is split into training, validation, and testing data according the number of event sequences, with each part accouting for 60%, 20% and 20%, respectively.

3.2 Compared Models and Implemented inf2vec

The compared models are state-of-the-art baselines. RNN based models include NHP (Mei & Eisner, 2017), FullyNN (Omi et al., 2019) and JTPP (Waghmare et al., 2022). Transformer based models include SAHP (Zhang et al., 2020a) and THP (Zuo et al., 2020). In the decoding stage, NHP, SAHP and THP model the intensity function, FullyNN models the cumulative hazard function and JTPP models the probability density function. We adapt JTPP to our framework, i.e., to implement inf2vec, we use RNN as our encoder, characterize the time distribution using probability density function, and train the model by maximum likelihood estimation. To reduce parameters, one single RNN ($Seq2vec$) is shared across different event types (Eq.4).

3.3 Prediction Results

The predictive ability of one NTPP model can be evaluated by predicting the next event, including the event type and occurrence time, given the historical events. As datasets used exhibit class (type) imbalance, we use weight F1 score to report the accuracy of event type prediction. For time prediction, we use mean absolute error as the evaluation metric. The results for the five datasets are summarized in Table 1 (Haw5 is not compared due to space limit). We can see that inf2vec outperforms the baselines in all datasets (except for JTPP). We use JTPP's encoder and decoder, and adapt them to our framework. We see the implemented inf2vec performs better than JTPP in most datasets. Compared with these baselines, our model conducts information decoupling in the view of event types, gaining more effectiveness in events embedding, encoding and decoding.

[1] https://x-datainitiative.github.io/tick/modules/hawkes.html

3.4 Type-Type Influences Learning

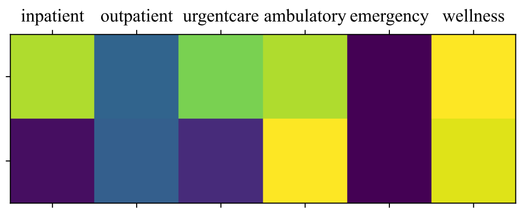

Figure 3: The illustration of our learned local embeddings over dataset SynEHR1. The first and second row show how other events influence ”wellness” and ”ambulatory”, respectively. Brighter color indicates stronger influence.

In Hawkes process (Eq.1), the coefficients $\alpha_{k,*}$ and $\beta_{k,*}$ indicate how other event types influence event type k, against which we can compare our learned type-type influences. In our framework inf2vec, the local embedding $\boldsymbol{z}_{\boldsymbol{m}}^{\boldsymbol{k}}(k_i)$ can naturally reflect the relationship between event type k_i and k. And in the vector space of event type k, close vector representations indicate that the corresponding event types have similar impacts on event type k. To evaluate the quality of type-type influences learning of our framework, for our learned local embeddings $\boldsymbol{z}_{\boldsymbol{m}}^{\boldsymbol{k}}(*) \in \mathbb{R}^d$ and the ground truth influences $[\alpha_{k,*}, \beta_{k,*}] \in \mathbb{R}^2$, we reduce the vector dimension to 1 by techniques like principle component analysis (PCA) Abdi & Williams (2010) and visualize them. Fig.2 illustrates the results on the three synthetic Hawkes datasets, showing that our learned local embeddings exhibit a high consistency with the ground truth influences, which validates the effectiveness of our proposed framework. Over EHR dataset SynEHR1, we show the learned event influences for event type ”wellness” and ”ambulatory” in Fig.3. The results show that our learned event influences are mostly consistent with human experience.

4 Conclusion

In this paper, we present a type-type influences learning framework Inf2vec for neural temporal point process, where the influences are directly parameterized and are learned end-to-end. Compared with conventional NTPP approaches, our framework conducts information decoupling from the perspective of event types, leading to more efficient embedding, encoding and decoding. Our framework is quite general for not posing any restriction on model’s encoder and decoder architecture. Experimental results on both synthetic and real-world EHR datasets demonstrate the superior performance of our model in terms of event prediction and influence learning task.

References

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31, 2018.

E. Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua A. Kulas, Andy Schuetz, and Walter F. Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Neural Information Processing Systems*, 2016. URL `https://api.semanticscholar.org/CorpusID:948039`.

Junyoung Chung, Çaglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *ArXiv*, abs/1412.3555, 2014.

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. A survey of the state of explainable ai for natural language processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pp. 447–459, 2020.

Saurabh Dash, Xueyuan She, and Saibal Mukhopadhyay. Learning point processes using recurrent graph network. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2022.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1555–1564, 2016.

Joseph Enguehard, Dan Busbridge, Adam James Bozson, Claire Woodcock, and Nils Y. Hammerla. Neural temporal point processes for modelling electronic health records. *ArXiv*, abs/2007.13794, 2020. URL `https://api.semanticscholar.org/CorpusID:220831434`.

Yulong Gu. Attentive neural point processes for event forecasting. In *AAAI Conference on Artificial Intelligence*, 2021. URL `https://api.semanticscholar.org/CorpusID:235306506`.

Alan G Hawkes. Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 33(3):438–443, 1971a.

Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971b.

Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2020.

B. Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. In *Knowledge Discovery and Data Mining*, 1998. URL `https://api.semanticscholar.org/CorpusID:232928`.

Bingqing Liu and Xikun Huang. Link-aware link prediction over temporal graph by pattern recognition. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 2023. URL `https://api.semanticscholar.org/CorpusID:260170307`.

Heikki Mannila, Hannu (TT) Toivonen, and A. Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1:259–289, 1997. URL `https://api.semanticscholar.org/CorpusID:6987161`.

Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in Neural Information Processing Systems*, 30, 2017.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26, 2013.

Dang Minh, H Xiang Wang, Y Fen Li, and Tan N Nguyen. Explainable artificial intelligence: a comprehensive review. *Artificial Intelligence Review*, pp. 1–66, 2022.

Takahiro Omi, Kazuyuki Aihara, et al. Fully neural network based model for general temporal point processes. *Advances in Neural Information Processing Systems*, 32, 2019.

Oleksandr Shchur, Ali Caner Türkmen, Tim Januschowski, and Stephan Günnemann. Neural temporal point processes: A review. *ArXiv*, abs/2104.03528, 2021.

Alexander Soen, Alexander Mathews, Daniel Grixti-Cheng, and Lexing Xie. Unipoint: Universally approximating point processes intensities. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9685–9694, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Govind Waghmare, Ankur Debnath, Siddhartha Asthana, and Aakarsh Malhotra. Modeling interdependence between time and mark in multivariate temporal point processes. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 1986–1995, 2022.

Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive hawkes process. In *International Conference on Machine Learning*, pp. 11183–11193. PMLR, 2020a.

W. Zhang, Thomas Kobber Panum, Somesh Jha, Prasad R. Chalasani, and David Page. Cause: Learning granger causality from event sequences using attribution methods. *Proceedings of the ... International Conference on Machine Learning. International Conference on Machine Learning*, 119:11235–11245, 2020b. URL `https://api.semanticscholar.org/CorpusID:211171699`.

Yunhao Zhang and Junchi Yan. Neural relation inference for multi-dimensional temporal point processes via message passing graph. In *International Joint Conference on Artificial Intelligence*, 2021. URL `https://api.semanticscholar.org/CorpusID:237100690`.

Ke Zhou, Hongyuan Zha, and Le Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *International Conference on Artificial Intelligence and Statistics*, 2013. URL `https://api.semanticscholar.org/CorpusID:8326502`.

Zihao Zhou and Rose Yu. Automatic integration for fast and interpretable neural point processes. In *Conference on Learning for Dynamics & Control*, 2023. URL `https://api.semanticscholar.org/CorpusID:259178556`.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *International Conference on Machine Learning*, pp. 11692–11702. PMLR, 2020.