Transformer Hawkes Process

Author: Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, Hongyuan Zha

February 13, 2023

MLE Reading Group

Introduction

Event Streams:

- Event happen at random times 0 < t_1 < t_2 …
- At time t_j, there occurs an event of type $k_j \in \{1, 2, \ldots, K\}$
- Format of data: S = $\{(t_j, k_j)\}_{j=1}^{L}$, where t_j is the time stamp, and k_j is the event type

Challenge of existing methods

- RNN based point process models fail to capture long-term dependencies
- Inputs are fed into the recurrent models sequentially, which means future states must be processed after the current state, rendering it impossible to process all the events in parallel

Introduction

Purpose:

- Transformer Hawkes Process is used to model event stream data
- Use the self-attention mechanism to capture the long-term dependencies and maintain computational efficiency

Goal:

- Predict the occurrence (when and what) of the next event

Background

Event Streams:

- Event happen at random times $0 < t_1 < t_2 \ldots$
- At time t_i, there occurs an event of type $k_i \in \{1, 2, \ldots, K\}$

Point Processes are generative models over event streams

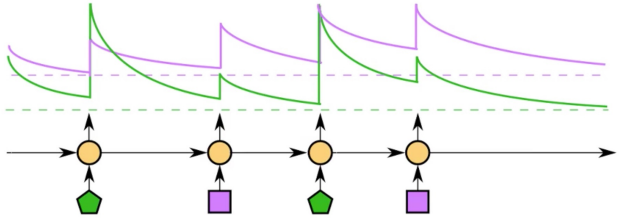
- Each event type has an intensity function $\lambda_k(t)$
- In $(t, t + dt]$ and event of type k occurs with probability $\lambda_k(t)dt$

Background

Hawkes Process (Hawkes, 1971) --- A self-Exciting Multivariate Point Process :

- Past event excite future events
- Intensity has a base level, plus excitation from each past event, which decays with time

$$\lambda(t) = \mu + \sum_{j:t_j<t} \psi(t - t_j).$$



Neural Hawkes Process (Mei & Eisner, 2017):

- Generalize the classical Hawkes process by parameterizing its intensity function with recurrent neural networks

Background

- **Masked Self-attention mechanism:**

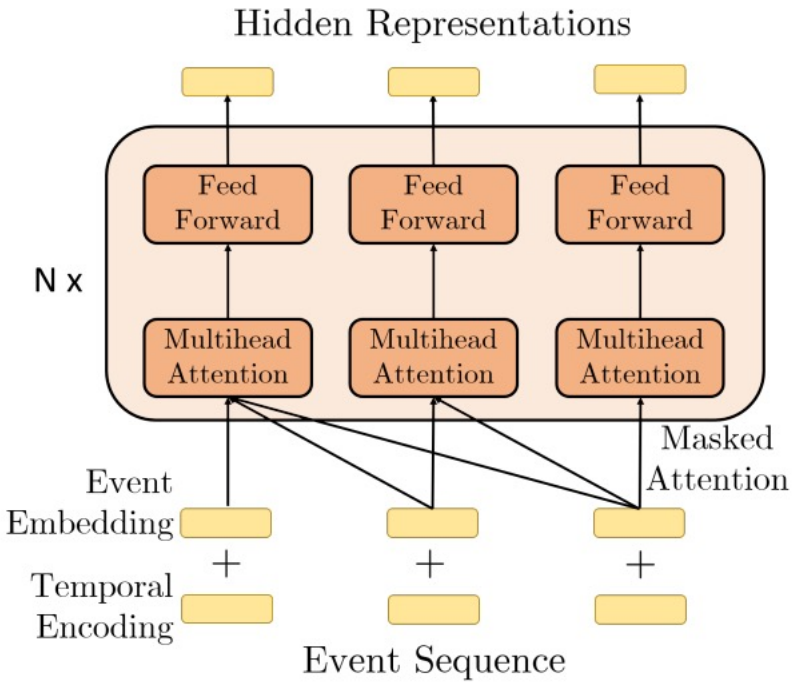
$$\mathbf{S} = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{M_K}}\right)\mathbf{V},$$
$$\mathbf{Q} = \mathbf{X}\mathbf{W}^Q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}^K, \quad \mathbf{V} = \mathbf{X}\mathbf{W}^V$$

  - Q --- Query; K --- Key; V --- Value
  - W --- Weight, learnable parameters
  - S is the attention output
  - To avoid “peeking into the future”, when computing the attention output S, mask all the future positions, i.e. Q(j, j+1)…Q(j, L) to inf

Model

- **Set up:** suppose we are given an event sequence S = $\{(t_j,k_j)\}_{i=1}^{L}$ of L events, where each event has type $k_j \in \{1,2,\dots,K\}$, with a total number of K types. Then each pair (t_j,k_j) corresponds to an event of type k_j occurs at time t_j

- Embedding layers contains a temporal encoding and an event embedding
- There are N layers of multi-head self-attention modules
- Each of the modules consists of a masked multi-head attention mechanism and a feed-forward neural network
- $h(t_j)$ encodes event (t_j,k_j) and its history

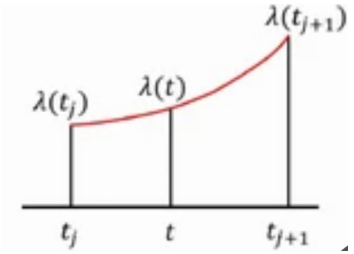


Model

- **Continuous Time Conditional Intensity**

$$\lambda(t|\mathcal{H}_t) = \sum_{k=1}^{K} \lambda_k(t|\mathcal{H}_t),$$

$$\lambda_k(t|\mathcal{H}_t) = f_k\Big(\underbrace{\alpha_k \frac{t - t_j}{t_j}}_{current} + \underbrace{\mathbf{w}_k^\top \mathbf{h}(t_j)}_{history} + \underbrace{b_k}_{base}\Big)$$



- $H_t = \{(t_j, k_j) : t_j < t\}$ is the history up to time t
- The "current" influence is an interpolation between two observed time stamps t_j and t_{j+1}

Training

- **Optimize model parameters**

Model parameters are learned by Maximizing the log-likelihood across all the sequences

$$\ell(\mathcal{S}) = \underbrace{\sum_{j=1}^{L} \log \lambda(t_j|\mathcal{H}_j)}_{\text{event log-likelihood}} - \underbrace{\int_{t_1}^{t_L} \lambda(t|\mathcal{H}_t)dt}_{\text{non-event log-likelihood}} .$$

Suppose we have N sequences $S_1, S_2, \ldots, S_N$, then the goal is to find parameters that solve $\max \sum_{i=1}^{N} \ell(\mathcal{S}_i)$,

This optimization problem can be efficiently solved by stochastic gradient type algorithms like ADAM

Non-event log-likelihood is approximated by Monte Carlo integration

Experimental Results

➢ **Baselines**

1. Recurrent Marked Temporal Point Process (RMTPP, Du et al. (2016))
2. Neural Hawkes Process (NHP, Mei & Eisner (2017))
3. Time Series Event Sequences (TSES, Xiao et al. (2017b))
4. Self-Attentive Hawkes Process (SAHP, Zhang et al. (2019))

➢ **Evaluation**

1. Per-event log-likelihood
2. Event type prediction accuracy
3. Event time prediction RMSE

Dataset	K	# Events	Avg. length
Retweets	3	2,173,533	109
MemeTrack	5000	123,639	3
Financial	2	414,800	2074
MIMIC-II	75	2,419	4
StackOverflow	22	480,413	72
911-Calls	3	290,293	403
Earthquake	2	256,932	500

Experimental Results

- **Log-Likelihood Comparison**

Model	RT	MT	FIN	MIMIC-II	SO
RMTPP	-5.99	-6.04	-3.89	-1.35	-2.60
NHP	-5.60	-6.23	-3.60	-1.38	-2.55
SAHP	-4.56	—	—	-0.52	-1.86
THP	**-2.04**	**0.68**	**-1.11**	**0.820**	**0.042**

- **Event Prediction Comparison**

Table 3. Event type prediction accuracy comparison.

Model	Financial	MIMIC-II	StackOverflow
RMTPP	61.95	81.2	45.9
NHP	62.20	83.2	46.3
TSES	62.17	83.0	46.2
THP	**62.64**	**85.3**	**47.0**

Table 4. Event time prediction RMSE comparison.

Model	Financial	MIMIC-II	StackOverflow
RMTPP	1.56	6.12	9.78
NHP	1.56	6.13	9.83
TSES	1.50	4.70	8.00
SAHP	—	3.89	5.57
THP	**0.93**	**0.82**	**4.99**

Experimental Results

- **Computational Efficiency**

Table 6. Sensitivity to the number of parameters and run-time comparison. Speedup is the speed of THP against NHP.

# Parameters	Log-likelihood		Speedup
	THP	NHP	
100k	−2.090	−6.019	×1.985
200k	−2.072	−5.595	×2.564
500k	−2.058	−5.590	×2.224
1000k	−2.060	−5.614	×1.778

Compare Neural Models for Hawkes Process

Different forms of dependency structures exist among past events and future events

- **Neural Hawkes Process**

  Author modeled a sequence of discrete events in continuous time by a novel ideal of continuous-time LSTM

- **Self-Attention Hawkes Process**

  Author examined the use of the self-attention mechanism in the Hawkes process by proposing a novel idea of a time-shifted positional embedding method, used non-linear transformation, dynamic decaying after the hidden representation to get the intensity function

- **Transformer Hawkes Process**

  Author used the self-attention mechanism for capturing the long-term dependencies and, at the same time, computationally very efficient compared to previous models

  Advantage: the self-attention mechanism model can directly select events whose occurrence time is at any distance from the current time

Conclusion

- Transformer Hawkes Process can capture both short-term and long-term dependencies
- Transformer Hawkes Process is computationally efficient

Worth Reading 

Implementation 

Thanks for listening!