Causality based event sequence modelling

Nikita Paplavskii [1] **Polina Pilyugina** [1]

Abstract

This project is a part of the Models of Sequential Data course and is dedicated to implementation of the Causal based Neural Hawkes model for event sequences. As a part of the project we will identify the most suitable method for creation of a causal graph for event sequences data. Further we introduce the causal graph into Neural Hawkes method using sparse matrices in order to improve the performance of the model. In the end, we will compare Causal based Neural Hawkes with original Neural Hawkes on real-worlds and synthetic data.

1. Introduction

The problem of event sequences modelling draws increased attention in recent years. Event sequences arise in many fields:

- *Banking*: events of customers' interactions with banking system. Analysis of such sequences allows to identify fraudulent behaviour or create individualized marketing products.
- *Medical events*: events of medical services usage of different patients. In this case, researchers are interested in exploration of sequences of events of other patients in order to predict other patients future medical occurrences.
- *Customer actions*: events of interaction of customers with some online services. In order to provide targeted marketing, researchers are interested in analysis of sequential patterns in customers' actions.
- *Recorded activity*: events recorded using smart wearable electronics and additional tracking apps. Patterns in human behaviour can make individualized recommendations easier.
- *Other events*: such as social media behaviour, web site activities etc.

With recent advances in area of Big Data gathering and storage, as well as increased computational capabilities, high-resolution event sequences data has become available for researchers. Among the main approaches for analysis of such type of data structure is using Hawkes process modelling (Hawkes, 1971). In recent years several approaches were introduced for Hawkes process modelling: parametric, such as (Xu et al., 2020), and neural based, such as (Mei & Eisner, 2017a) and (Zuo et al., 2021). Although neural methods provide promising results, this approaches usually do not take into account internal causal relationships. Causality defines inter dependencies between time series. Usually, casual relationship can be represented using graph structure, or corresponding sparse adjacency matrix. One can possibly increase the model performance, by taking into account existence or non-existence of causal relationship between series. Moreover, such approach can allow for improved computational difficulty using sparse matrices for causality relationship. This project is dedicated to incorporation of casual adjacency matrix into Neural Hawkes model and evaluation of the results.

This paper has the following structure: in 2 we define the problem and motivation behind the research. In section 3 we identify main existing approaches which we use as a base of our work. Further in section 3.1 we identify main methods we use in our implementation. The code structure and its main features are described in 4. Finally, we will provide results of our experiments in 7.

2. Problem statement

Given K types of possible events, the event sequence E is defined as

$$E = ((k_1, t_1), (k_2, t_2), ..., (k_N, t_N)) \qquad (1)$$

where N is total number of events in sequence, $k \in \{1, 2, ..., K\}$ is an event type that occurred and $t_1 < t_2 < ... < t_N$ is continuous time of event occurrence. The problem of predicting the future of a sequence means identifying which events are likely to happen next and when they will happen. This problem is different from common time series

[*]Equal contribution [1]Skolkovo Institute of Science and Technology, Moscow, Russia. Correspondence to: Nikita Paplavskii <Nikita.Paplavsky@skoltech.ru>.

Final Projects of the Machine Learning 2020 Course, Skoltech, Moscow, Russian Federation, 2020.

forecasting, since the event sequence is in continuous time and no fixed time-lagged observation is available Moreover, events are discrete and sparse, which even more complexifies the problem. Therefore common methods for time series analysis are not of particular use when dealing with event sequences.

For the problem of event sequences prediction neural approach called Neural Hawkes model was introduced. It allowed for significant improvement over other methods, as will be discussed in 3.1. Although Neural Hawkes provides good results, it is prone to several complexities. Firstly, Neural Hawkes training is computationally intense. For event sequence modelling we use large amounts of data, therefore an ability to improve computational complexity via sparcification can possibly improve computational difficulty. Moreover, authors identify that an important extension to the model would be adding information on causal relationship between event sequences. Although the model can identify the dependency between events, it is important to use non symmetrical causal relationship between series in order to improve the performance. Moreover, the usage of a sparse Causal adjacency matrix will allow to easier introduce sparcification into the model.

Many recent works outline importance of Granger Causality graphs for time series data analysis, such as (Arnold et al., 2007) and (Basu et al., 2015). However, their approaches are based mostly on vector auto-regressive model for discrete time-lagged variables. This approach is useful for time series, but is not directly applicable to continuous-time event sequences data. Therefore, other approaches arose for modelling Granger Causality on event sequences. Among them is work of (Xu et al., 2016) which proposes an approach to calculate Ganger Causality for Hawkes process. However, this approach was not yet introduced into the Neural Hawkes model, and in this project we aim to cover this topic.

3. Related Works

3.1. Methods

In this work we will use Neural Hawkes Process and Granger causality.

The original Neural Hawkes Process model is introduced in (Mei & Eisner, 2017b). Authors suggested a method to improve Hawkes model's expressiveness (Hawkes, 1971). Original Hawkes model describes a set of event intensities as:

$$\lambda_k(t) = \mu_k + \sum_{h:t_h<t} \alpha_{k_h,k} \text{exp}(-\delta_{k_h,k}(t-t_h)) \tag{2}$$

where $\mu_k \geq 0$ is the base intensity of event type k, $\alpha_{j,k} \geq 0$ is the degree to which an event of type j initially excites type k, and $\delta_{j,k} \geq 0$ is the decay rate of that excitation. One constraint of this approach is that the positivity constraints in the Hawkes process limit its expressivity. First, the positive interaction parameters $\alpha_{j,k}$ fail to capture inhibition effects, in which past events reduce the intensity of future events. Second, the positive base rates μ_k fail to capture the inherent inertia of some events, which are unlikely until their cumulative excitation by past events crosses some threshold. To remove such limitations, authors suggested two self-modulating models.

As the first step they relaxed positivity constraints over $\alpha_{j,k}$ and μ_k allowing them to range over R , which allows inhibition ($\alpha_{j,k} < 0$) and inertia ($\mu_k < 0$). To solve possible negativity of resulting total activation they pass through a non-linear transfer function $f : R \to R^+$ to obtain a positive intensity function as required:

$$\begin{aligned} \tilde{\lambda}_k(t) &= \mu_k + \sum_{h:t_h<t} \alpha_{k_h,k} \exp(-\delta_{k_h,k}(t-t_h)) \\ \lambda_k(t) &= f_k(\tilde{\lambda}_k(t)) \end{aligned} \tag{3}$$

As non-linear transfer function authors used scaled softplus function. Second step two improve expressiveness of Hawkes model is using recurrent network to deal with more complex dependencies instead of assumption that previous events have independent, additive influence on $\tilde{\lambda}_k(t)$.

Neural Hawkes model provides good results, but according to authors performance can be improved further if model will use causal relationships. A model that can predict the result of interventions is called a causal model.

In this work we will use Ganger causality. Ganger causality is a method to evaluate causality between processes. We can describe it as follow:

- Let X and Y be stationary stochastic processes.
- Denote with $\mathcal{U}_i = (U_{i-1}, ..., U_{i-\infty})$ all the information in the universe until time i, and with $\mathcal{X} = (X_{i-1}, ..., X_{i-\infty})$ all information in X until time i.
- Denote with $\sigma^2(Y_i|\mathcal{U}_i)$ the variance of the residual of predicting Y_i using $\mathcal{U}$ at time i.
- Denote with $\sigma^2(Y_i|\mathcal{U}_i \setminus \mathcal{X}_i)$ the variance of the residual of predicting Y_i using all information in $\mathcal{U}_i$ at time i except $\mathcal{X}_i$.

If $\sigma^2(Y_i|\mathcal{U}_i) < \sigma^2(Y_i|\mathcal{U}_i \setminus \mathcal{X}_i)$ then we say that X Granger-causes Y, and write X⇒Y If X⇒Y and Y⇒X we say that feedback is occurring. Therefore, we can create non-symmetric adjacency matrix of causal relationship between events.

4. Implementation

For the code, we will base our work on some existing available implementations. Firstly, existing implementation of Neural Hawkes on PyTorch provides us with a base Continious Time Long-Short Term Memory (CTLSTM) model (Mei & Eisner, 2017b). We aim to improve the model in order to make use of sparse adjacency causal graph matrix and thus sparcify the network. We also aim to make some improvement to the code in order to make it more flexible. Secondly, we found an existing implementation of Granger Causality for Hawkes process in MatLab (Xu & Zha, 2017). We aim to firstly check this implementation and the math behind it. Further we will rewrite it in python and improve it using sparse matrices. Moreover, we aim to use another libraries with similar work, such as *tick* in order to check our results.

In order to do experiments we aim to use the *tick* library for synthetic Neural Hawkes process simulations. This will allow us additionally to check our algorithm for Granger Causality for Hawkes process calculation.

The code on our implementation will be provided on GitHub repository.

5. Our approach

In general, as we discussed in previous sections, we aim to introduce Causality matrix into Neural Hawkes in order to improve its performance and make use of causal relationships between event sequences. Firstly, we implement the method for Granger Causality calculation as described in 3.1. This will allow us to create a sparse adjacency matrix of causal graph. Further, we aim to include this matrix into the model in order to allow the model to get use of causal relationships. Our initial idea is to use sparse causal matrix to intensify or suppress certain events.

6. Baseline

As for the baseline, we will use Neural Hawkes implementation described in 4. In our opinion, this would be the best choice as we will base our implementation on this model.

7. Results

7.1. Experimental setup

For datasets we will use both synthetic and real-world data. As for real-world data, we will obtain it from the original Neural Hawkes implementation on the GitHub. Description of these datasets is contained in 1. For the purpose of our experiments we will focus on the data with large number of event types, i.e. larger than 2.

Dataset	Number of event types
SYNTHETIC	5
RETWEETS	3
MEMETRACK	5000
MIMIC-II	75
STACKOVERFLOW	22
FINANCIAL	2

Table 1. Description of datasets from Neural Hawkes model implementation

As for synthetic dataset, we will rely on *tick*[1] library

8. Conclusion

In this section we will make conclusions based on the results in 7.

References

Arnold, A., Liu, Y., and Abe, N. Temporal causal modeling with graphical granger methods. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 66–75, 2007. doi: 10.1145/1281192.1281203.

Basu, S., Shojaie, A., and Michailidis, G. Network granger causality with inherent grouping structure. *Journal of Machine Learning Research*, 16(13):417–453, 2015. URL `http://jmlr.org/papers/v16/basu15a.html`.

Hawkes, A. G. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971. ISSN 00063444. URL `http://www.jstor.org/stable/2334319`.

Mei, H. and Eisner, J. The neural hawkes process: A neurally self-modulating multivariate point process, 2017a.

Mei, H. and Eisner, J. The neural Hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*, Long Beach, December 2017b. URL `https://arxiv.org/abs/1612.09328`.

Xu, G., Wang, M., Bian, J., Huang, H., Burch, T. R., Andrade, S. C., Zhang, J., and Guan, Y. Semi-parametric learning of structured temporal point processes. *Journal of Machine Learning Research*, 21(192):1–39, 2020. URL `http://jmlr.org/papers/v21/18-735.html`.

[1] `https://x-datainitiative.github.io/tick/index.html`

Xu, H. and Zha, H. Thap: A matlab toolkit for learning with hawkes processes, 2017.

Xu, H., Farajtabar, M., and Zha, H. Learning granger causality for hawkes processes, 2016.

Zuo, S., Jiang, H., Li, Z., Zhao, T., and Zha, H. Transformer hawkes process, 2021.

A. Team member's contributions

Explicitly stated contributions of each team member to the final project.

Nikita Paplavskii (50% of work)

- Review of literature on Granger Causality for event sequences;
- Implementing Granger Causality for event sequences in python;
- Running experiments with Granger Causality algorithm;
- Review of the datasets for experiments;

Polina Pilyugina (50% of work)

- Review of literature on sparse Neural Networks and Neural Hawkes;
- Integration of Granger Causality into Neural Hawkes;
- Synthetic datasets creation for experiments;
- Experiments with Causal based Neural Hawkes;

B. Reproducibility checklist

Answer the questions of following reproducibility checklist. If necessary, you may leave a comment.

1. A ready code was used in this project, e.g. for replication project the code from the corresponding paper was used.

   ☐ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment:

2. A clear description of the mathematical setting, algorithm, and/or model is included in the report.

   ☐ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment:

3. A link to a downloadable source code, with specification of all dependencies, including external libraries is included in the report.

   ☐ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment:

4. A complete description of the data collection process, including sample size, is included in the report.

   ☐ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment:

5. A link to a downloadable version of the dataset or simulation environment is included in the report.

   ☐ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment:

6. An explanation of any data that were excluded, description of any pre-processing step are included in the report.

   ☐ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment:

7. An explanation of how samples were allocated for training, validation and testing is included in the report.

   ☐ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment:

8. The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results are included in the report.

   ☐ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment:

9. The exact number of evaluation runs is included.

   ☐ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment:

10. A description of how experiments have been conducted is included.

    ☐ Yes.
    ☐ No.
    ☐ Not applicable.

    Students' comment:

11. A clear definition of the specific measure or statistics used to report results is included in the report.

    ☐ Yes.
    ☐ No.
    ☐ Not applicable.

    Students' comment:

12. A description of the computing infrastructure used is included in the report.

    ☐ Yes.
    ☐ No.
    ☐ Not applicable.

    Students' comment: