Accelerating Hawkes Process for Modelling Event History Data

Ashwin Ram [1] **P. K. Srijith** [2]

Abstract

Hawkes Processes are probabilistic models useful for modelling the occurrences of events over time. They exhibit mutual excitation property, where a past event influences future events. This has been successful in modelling the evolution of memes and user behaviour in social networks. In the Hawkes process, the occurrences of events are determined by an underlying intensity function which considers the influence from past events. The intensity function models the mutual-exciting nature by adding up the influence from past events. The calculation of the intensity function for every new event requires time proportional to the number of past events. When the number of events is high, the repeated intensity function calculation will become expensive. We develop a faster approach which takes only constant time complexity to calculate the intensity function for every new event in a mutually exciting Hawkes process. This is achieved by developing a recursive formulation for mutually exciting Hawkes process and maintaining an additional data structure which takes a constant space. We found considerable improvement in runtime performance of the Hawkes process applied to the sequential stance classification task on synthetic and real world datasets.

1. Introduction

Applications involving event history data analysis arise in various domains such as healthcare, social networks, and the Web. For instance, in online social networks such as Twitter one might be interested in knowing the time at which a user tweets about a topic. In recommendation systems, companies will be interested in knowing the time at which a user buys an item. In these tasks, point processes have been found to be useful to model the occurrence of events. They are characterized by an intensity function which specifies a distribution over the events occurring in some time interval.

In many cases, events are associated with some markers which denote the category associated with the event. The events in one category can cause more events from that category or other categories resulting in a cascade (e.g. in Twitter posts from an user could influence posts from other users). Hawkes process (HPs) with mutual excitation have been found to be useful to model such influences among the events. They define the intensity function to be a function of past events and their categories. Every event which has occurred in the past will have a positive influence which decays exponentially over time and is weighted by the category specific mutual influences. The intensity function sums up influence from past events and this becomes computationally expensive as more and more events happens. The intensity function needs to be recalculated for every data point and the complexity over all the N events becomes $\mathcal{O}(N^2)$. This makes both the training and prediction time expensive. We develop a recursive algorithm which could speed-up the intensity function calculation for mutually exciting Hawkes process (*accelerated Hawkes*), resulting in constant time computation of the intensity function for an event. This reduces the complexity of computing the intensity function over all the data points to $\mathcal{O}(N)$, leading to faster training and inference of the Hawkes process models.

We consider a Hawkes process model with mutual excitation developed for the stance classification problem involving temporal textual data (Lukasik et al., 2016). Here we treat text classification as a sequence labelling problem, also taking into account the times associated with the labels corresponding to past text. For instance, we would like to classify tweets into different categories by considering the categories associated with past tweets and their times. We develop the accelerated Hawkes process variant of this model, and show the runtime performance improvement on synthetic and real world data sets arising in the web such as Twitter and Amazon review data.

[1] National Institute of Technology Trichy, India [2] Indian Institute of Technology Hyderabad, India. Correspondence to: P. K. Srijith <srijith@iith.ac.in>.

ICML 2017 Time Series Workshop, Sydney, Australia, 2017. Copyright 2017 by the author(s).

2. Related Work

Multiple works modeled occurrences of posts in social media platforms using mutually exciting Hawkes Processes (HP).(Yang & Zha, 2013) employed HP for inferring the underlying network of connections of users based on the observations of timestamps and text content of tweets. Also, prior knowledge about the users and connections from a social network has been incorporated into the Hawkes Process model (Zhou et al., 2013; Kobayashi & Lambiotte, 2016; Srijith et al., 2017). They have been used to model the generation of tweets over a continuous time domain (Zhao et al., 2015) and stance classification of tweets (Lukasik et al., 2016). Joint modeling of information spread and text has been considered by (He et al., 2015), who introduced a joint model of topics and network inference from information propagation. (Du et al., 2015) introduced a Dirichlet-Hawkes Process model, which models clustering of events across Hawkes processes via a Dirichlet process. Hawkes process have also found its application healthcare for disease progression modelling (Choi et al., 2015). All of these application would benefit from the proposed approach which will speed up intensity function calculations. A recursive algorithm for self exciting Hawkes process is provided in (Laub et al., 2015). Here, we develop a recursive algorithm for mutually exciting Hawkes process.

3. Time Sensitive Classification of Events

We consider N events where each event is represented as a tuple (t_n, x_n, y_n), where t_n denotes the time at which an event occurs, x_n is the textual content, and y_n is the label. We consider the task of classifying each event to a label category $y_n \in \{1, 2, \dots S\}$ by considering the content x_n, and the times and labels associated with past events. We devise efficient Hawkes process approach to perform this task of time sensitive sequence classification.

4. Hawkes Processes

Point processes are useful for modelling longitudinal data. They are characterized by an intensity function $\lambda(t|H_t) > 0$ (the conditioning is on history of events until time t) which provides the instantaneous probability of occurrence of an event at some time t. One example of a point process is a Hawkes process (HP), which models the intensity function by adding up influence from past events. The intensity function associated with a mutually exciting Hawkes process takes the following form:

$$\lambda_{y_n}(t) = \mu_{y_n} + \sum_{t_l < t} \alpha_{y_l, y_n} \kappa(t - t_l) \qquad (1)$$

where the first term represents the constant base intensity of generating label y. The second term represents the influence from the events that happen prior to time of interest. The influence from each event decays over time and is modelled using an exponential decay term $\kappa(t - t_l) = exp(-\omega(t - t_l))$. The matrix α of size $S \times S$ encodes the degrees of influence between pairs of labels assigned to the events. In the Hawkes process model used for stance classification (Lukasik et al., 2016), the intensity is multiplied by likelihood of generating the context (text or other features) given the label. This is modelled as a multinomial distribution conditioned on the label,

$$p(x_n|y_n) = \prod_{v=1}^{V} \beta_{y_n v}^{x_{nv}}, \qquad (2)$$

where V is the feature size and β is the matrix of size $S \times V$ specifying the probability distribution over features for every label. The parameters of the model are learnt by maximizing the log-likelihood of observing the text, labels and times at which they occur in the data set.

$$l(\mu, \alpha, \omega, \beta) = -\sum_{y=1}^{S} \int_0^T \lambda_y(t) dt + \sum_{n=1}^{N} \log \lambda_{y_n}(t_n) + \sum_{n=1}^{N} \sum_{v=1}^{V} x_{nv} \log \beta_{y_n v}. \qquad (3)$$

For every new label, the summation in (1) has to be computed over all the previous data points and this makes it computationally inefficient. For a total of N events, the complexity of intensity function calculation is $\frac{N(N-1)}{2}$, *i.e.* $\mathcal{O}(N^2)$. We consider a method to overcome this computational overhead and enable Hawkes process to be applicable to large datasets.

4.1. Accelerated Hawkes Process

Consider a sequence of labels $[y_1, y_2, \dots, y_n]$ and times $[t_1, t_2, \dots, t_n]$. First, we initialise a S dimensional vector $\gamma(t)$ at time t_1 as,

$$\gamma(t_1) = \begin{bmatrix} \alpha_{y_1,1} exp(\omega t_1) \\ \alpha_{y_1,2} exp(\omega t_1) \\ \vdots \\ \alpha_{y_1,S} exp(\omega t_1) \end{bmatrix} \qquad (4)$$

Now to calculate $\sum_{t_l < t_2} \alpha_{y_l, y_2} \kappa(t_2 - t_l) = \alpha_{y_1, y_2} \kappa(t_2 - t_1) = \alpha_{y_1, y_2} exp(-\omega(t_2 - t_1))$, we multiply $exp(-\omega t_2)$ to the row corresponding to the label y_2 in γ. Following this we update the state of the vector γ for intensity function calculation in the next point. Vector γ at time t_2 is updated by considering the label of y_2 and adding the term $\alpha_{y_2,.}exp(\omega t_2)$ to each row.

$$\gamma(t_2) = \begin{bmatrix} \alpha_{y_1,1}exp(\omega t_1) + \alpha_{y_2,1}exp(\omega t_2) \\ \alpha_{y_1,2}exp(\omega t_1) + \alpha_{y_2,2}exp(\omega t_2) \\ \vdots \\ \alpha_{y_1,S}exp(\omega t_1) + \alpha_{y_2,S}exp(\omega t_2) \end{bmatrix} \tag{5}$$

Similarly,to calculate $\sum_{t_l<t_3} \alpha_{y_l,y_3}\kappa(t_3 - t_l)$, we multiply $exp(-\omega t_3)$to the row corresponding to the label y_3.This can be extended to n points to give $\gamma(t_n)$ as,

$$\gamma(t_n) = \begin{bmatrix} \sum_{i=1}^{n} \alpha_{y_i,1}exp(\omega t_i) \\ \sum_{i=1}^{n} \alpha_{y_i,2}exp(\omega t_i) \\ \vdots \\ \sum_{i=1}^{n} \alpha_{y_i,S}exp(\omega t_i) \end{bmatrix} \tag{6}$$

Thus in each step we have 4 sums to compute, giving a total complexity of $\mathcal{O}(N)$ for N points.

The intensity function can be written recursively with the help of $\gamma(t_n)$. The intensity function for the event $(n + 1)$ is,

$$\lambda_{y_{n+1}} = \mu_{y_{n+1}} + exp(-\omega t_{n+1})\gamma(t_n)_{y_{n+1}} \tag{7}$$

and the intensity function for the event n is,

$$\lambda_{y_n} = \mu_{y_n} + exp(-\omega t_n)\gamma(t_{n-1})_{y_n} \tag{8}$$

where, $\gamma(t_n)_{y_{n+1}}$ represents the y_{n+1} row of $\gamma(t_n)$. From (6), we know that

$$\gamma(t_{n-1})_{y_n} = \gamma(t_n)_{y_n} - \alpha_{y_n,y_n}exp(\omega t_{n-1}) \tag{9}$$

Combining (7), (8) and (9), the intensity function can be written recursively as

$$\begin{aligned} \lambda_{y_{n+1}} = \mu_{y_{n+1}} &+ [\gamma(t_n)_{y_{n+1}} - \gamma(t_n)_{y_n}]exp(-\omega t_{n+1}) \\ &+(\lambda_{y_n} - \mu_{y_n} + \alpha_{y_n,y_n})\kappa(t_{n+1} - t_n) \end{aligned} \tag{10}$$

5. Experiments and Results

We conduct experiments on synthetic and real world data sets from Twitter and product reviews. We compare the computational time required by the standard implementation of the Hawkes process and the proposed approach. All the experiments are run on a machine with 2.9 GHz Intel Core $i5$ processor and 8 GB RAM.

Algorithm 1: Ogata's Thinning algorithm

```
Input: conditional intensity function $\lambda_y(t)$, time $T$
$t = 0, S = \{\}$.
while $t < T$ do
    $\beta \leftarrow \sum_{y=1}^{S} \lambda_y(t)$
    Generate candidate next arrival time from
    $s \sim \exp(1/\beta)$
    Generate random number $U \sim Unif([0, 1])$
    if $(t + s > T)$ OR $(U > \frac{\sum_{y=1}^{S} \lambda_y(t+s)}{\beta})$ then
        Set $t = t + s$
    else
        Set $t = t + s$
        Obtain $\delta_y = \frac{\lambda_y(t)}{\sum_{y=1}^{S} \lambda_y(t)} \forall y = 1 \dots S$
        Sample label $l$ from $Cat(\delta_1, \dots, \delta_S)$
        $S = S \cup (t, l)$, Update Intensity $\lambda_y(t)$
    end if
end while
Return: $S$
```

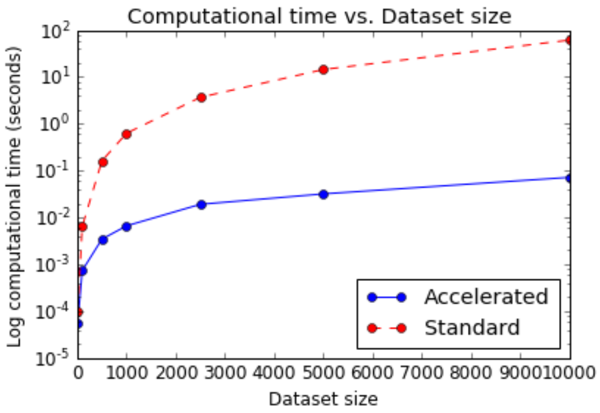

Figure 1. Time comparison for Standard vs. accelerated Hawkes for synthetic data

5.1. Performance on Synthetic Data

The synthetic data set consisting of time and labels of events is generated using Ogata's Thinning Algorithm (Ogata, 1981). The algorithm iteratively samples time from an exponential distribution with scale parameter β to be the inverse of the summation of intensities over all labels. The sample is accepted based on the ratio of intensities summed over labels at the new point and the previous point. The sampled time is used to compute the new intensity across each label and summation of intensities over all the labels. The label is sampled from a categorical distribution with each label having a probability given by the ratio of its current intensity to the summed intensity. Algorithm 1 outlines the data generation process. This is equivalent to sampling from a multi-variate Hawkes process. The

α matrix, μ and ω parameters are randomly initialized and the initial time is chosen to be zero. For this synthetic data set, we consider the computational time required for intensity calculation using both methods.

From Figure 1 it can be seen that the proposed model achieved much better performance in computational time. Time increased quadratically with increasing data set size for the standard model. We see that with for every tenfold increase in dataset set size, there is a hundred times increase in computing time. The proposed approach achieved close to three orders of magnitude improvement in computational time compared to the standard model on 5000 data points. This will enables the use of Hawkes process to the problems involving large event history data.

5.2. Performance on Ferguson Twitter Dataset

To validate the computational improvement further, we apply it to the rumour stance classification problem in (Lukasik et al., 2016). We tested the Accelerated HP approach for the Ferguson Twitter data set. The dataset consists of 1244 data points. It consists of tweet time, meme id (conversation thread id), infecting id (reply to id), labels and the tweet message.

The stance classification task involves each tweet d_j being classified into one of the four categories $y_j \in Y$, which represents the stance of the tweet d_j with respect to the rumour R_i it belongs to. Four categories are considered in the dataset ; support, deny, question, comment. We consider the leave-one-out (LOO) setting, introduced by Lukasik et al. (2015a), where for each rumour $R_i \in D$ we construct the test set R_i and the training set D/R_i. In each fold i, the HP parameters are learnt from the training set and tested on R_i. For this setting, we compare both the approaches on the training time taken to learn the parameters of the model over all the folds.

Table 1. Training time comparison for standard and acclerated Hawkes for Fergson Twitter data

Data set size	Accelerated (sec)	Standard (sec)
10	0.022	0.062
100	0.270	1.606
1000	20.965	110.860

From Table 1 it can be seen that proposed approach lead to faster training time than the standard approach. For 1000 data points the proposed approach is found to be 5 times faster than the standard approach.

5.3. Performance on Amazon review Dataset

We further consider the performance of the accelerated Hawkes method for a larger dataset. The Amazon instant video review dataset (He & McAuley, 2016) consists of 37,216 reviews. This datasets consists of review post time, product id, rating, review text. We apply a Hawkes process to model the prediction of the rating class of a product similarly as in the rumour stance classification in the previous section.

We first pre-process the data by converting the overall rating into a set of three classes which include bad, good and excellent by thresholding. Also, since the review text columns contain a lot of symbols, spelling errors and named entities we remove the stop words and use a lemmatizer and the Synsets from Wordnet to get the root word and check for word existence respectively. The processed vocabulary is then used to convert the text into a one hot encoded vector based on the collected vocabulary. We further used only the 100 most reviewed products with around $15K$ points and followed the leave-one-out approach described in the earlier section. Here, we iteratively leave each product out, train the HP model on the remaining products and predict on the leaved out product. We see a stark decrease in training time using accelerated Hawkes over standard Hawkes. Accelerated Hawkes took only 1.46 hours, while standard Hawkes took 11 hours to learn the parameters of the Hawkes process model over all the 100 folds. Both the approaches gave the same accuracy of 0.65.

Table 2. Comparison of Training time for Standard vs. Acclerated Hawkes on Amazon review data

Method	Training Time
Accelerated	5263.47 sec (1.46 hours)
Standard	39610.17 sec (11 hours)

6. Conclusions

In this paper we considered an efficient approach to Hawkes process intensity function calculation and demonstrated practical gains on one synthetic and two real world datasets. By maintaining a constant size vector, the proposed Hawkes process approach managed to reduce the time complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. This will enable Hawkes processes to be applied to the problems involving big data. Though the performance improvement is shown on the stance classification task using Hawkes process, the proposed approach is generic and will be useful for many tasks involving Hawkes process.

7. Acknowledgements

We thank Dr. Trevor Cohn and Michal Lukasik for inspiring discussions on Hawkes process.

References

Choi, Edward, Du, Nan, Chen, Robert, Song, Le, and Sun, Jimeng. Constructing disease network and temporal progression model via context-sensitive hawkes process. In *IEEE International Conference on Data Mining, ICDM*, pp. 721–726, 2015.

Du, Nan, Farajtabar, Mehrdad, Ahmed, Amr, Smola, Alexander J., and Song, Le. Dirichlet-Hawkes Processes with Applications to Clustering Continuous-Time Document Streams. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pp. 219–228, 2015.

He, Ruining and McAuley, Julian. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web, WWW*, pp. 507–517, 2016.

He, Xinran, Rekatsinas, Theodoros, Foulds, James, Getoor, Lise, and Liu, Yan. HawkesTopic: A Joint Model for Network Inference and Topic Modeling from Text-Based Cascades. *ICML*, 2015.

Kobayashi, Ryota and Lambiotte, Renaud. Tideh: Time-dependent hawkes process for predicting retweet dynamics. *CoRR*, abs/1603.09449, 2016.

Laub, Patrick J., Thomas, Taimre, and Philip K., Pollett. Hawkes processes. In *arXiv preprint arXiv:1507.02822*, 2015.

Lukasik, Michal, Srijith, P. K., Vu, Duy, Bontcheva, Kalina, Zubiaga, Arkaitz, and Cohn, Trevor. Hawkes processes for continuous time sequence classification: an application to rumour stance classification in twitter. In *ACL*, 2016.

Ogata, Yosihiko. On lewis' simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1):23–30, 1981.

Srijith, P. K., Lukasik, Michal, Bontcheva, Kalina, and Cohn, Trevor. Longitudinal Modeling of Social Media with Hawkes Process based on Users and Networks. In *ASONAM*, 2017.

Yang, Shuang-Hong and Zha, Hongyuan. Mixture of mutually exciting processes for viral diffusion. In *ICML (2)*, volume 28, pp. 1–9, 2013.

Zhao, Qingyuan, Erdogdu, Murat A., He, Hera Y., Rajaraman, Anand, and Leskovec, Jure. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proc. of International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1513–1522, 2015.

Zhou, Ke, Zha, Hongyuan, and Song, Le. Learning social infectivity in sparse low-rank networks using multidimensional hawkes processes. In *AISTATS*, volume 31, pp. 641–649, 2013.