

Adversarial Self-Paced Learning for Mixture Models of Hawkes Processes

Dixin Luo¹ Hongteng Xu^{2,1} Lawrence Carin¹

Abstract

We propose a novel adversarial learning strategy for mixture models of Hawkes processes, leveraging data augmentation techniques of Hawkes process in the framework of self-paced learning. Instead of learning a mixture model directly from a set of event sequences drawn from different Hawkes processes, the proposed method learns the target model iteratively, which generates “easy” sequences and uses them in an adversarial and self-paced manner. In each iteration, we first generate a set of augmented sequences from original observed sequences. Based on the fact that an easy sample of the target model can be an adversarial sample of a misspecified model, we apply a maximum likelihood estimation with an adversarial self-paced mechanism. In this manner the target model is updated, and the augmented sequences that obey it are employed for the next learning iteration. Experimental results show that the proposed method outperforms traditional methods consistently.

1. Introduction

Real-world event sequences are often modeled based on temporal point processes. Specifically, a temporal point process with C event types can be represented as a counting process $N(t) = \{N_c(t)\}_{c \in \mathcal{C}}$, where each $N_c(t)$ is the number of type- c events happening at or before time t and $\mathcal{C} = \{1, \dots, C\}$. As a special kind of point process, Hawkes process (Hawkes, 1971) formulates the expected instantaneous happening rate of type- c events, or called *intensity function*, as

$$\lambda_c(t) = \frac{\mathbb{E}[dN_c(t)|\mathcal{H}^c(t)]}{dt} = \mu_c + \sum_{i:t_i < t} \phi_{cc_i}(t - t_i),$$

where $\mathcal{H}^c(t) = \{(t_i, c_i) | t_i < t, c_i \in \mathcal{C}\}$ contains historical events before time t , μ_c is the base intensity capturing

exogenous fluctuations of the type- c event, and $\phi_{cc'}(t)$ is the impact function measuring the infectivity of the type- c' event to the type- c event type over time. Therefore, we denote an event sequence yielding to a Hawkes process as $\mathbf{s} \sim \text{HP}(\boldsymbol{\mu}, \Phi)$, with basic intensity $\boldsymbol{\mu} = [\mu_c] \in \mathbb{R}^C$ and impact functions $\Phi = [\phi_{cc'}^k(t)]$.

As an extension of Hawkes process, the mixture model of Hawkes processes (MixHP) is capable of describing the clustering structure of different event sequences and capturing the dependency among events within each cluster. MixHP has been used to model real-world event sequences, *e.g.*, patient admissions (Xu & Zha, 2017), social behaviors (Yang & Zha, 2013), and user logs of information systems (Luo et al., 2015). Suppose that the sequences in $\mathcal{S} = \{\mathbf{s}_n = (t_i^n, c_i^n)_{i=1}^{I_n}\}_{n=1}^N$ are generated via K different Hawkes processes, *i.e.*,

$$k \sim \boldsymbol{\pi}, \quad \mathbf{s}|k \sim \text{HP}(\boldsymbol{\mu}^k, \Phi^k), \quad \text{for } \mathbf{s} \in \mathcal{S}, \quad (1)$$

where $\boldsymbol{\pi} = [\pi_k] \in \Sigma^K$ represents the distribution of the K Hawkes processes. Accordingly, the likelihood of a sequence \mathbf{s} is represented as

$$p(\mathbf{s}; \Theta) = \sum_{k=1}^K \pi_k p(\mathbf{s} | \boldsymbol{\mu}^k, \Phi^k). \quad (2)$$

Here, $p(\mathbf{s} | \boldsymbol{\mu}^k, \Phi^k) = \prod_{i=1}^I \lambda_{c_i}^k(t_i) e^{-\sum_c \int_0^T \lambda_c^k(s) ds}$ is the likelihood of the sequence \mathbf{s} conditioned on the k -th Hawkes process $\text{HP}(\boldsymbol{\mu}^k, \Phi^k)$.

Given observed sequences, we can apply maximum likelihood estimation (MLE) to learn the target mixture model, as shown in Figure 1(a). However, in practice this learning strategy often suffers from insufficient data. For example, in the admission record dataset MIMIC-III (Johnson et al., 2016), most patients only have 5 admissions or fewer in ten years, while there are over 600 kinds of diseases (*i.e.*, types of events). Learning from such short sequences leads to serious over-fitting. For a single Hawkes process, such a problem can be mitigated by various data augmentation techniques, *e.g.*, randomly stitching (Xu et al., 2017) or superposing (Xu et al., 2018b) the original sequences. Unfortunately, for mixture models of Hawkes processes, these techniques cannot be applied directly, because in general the stitching/superposing result of two sequences from different clusters does not yield to a Hawkes process, which may cause serious model misspecification.

¹Department of ECE, Duke University, Durham, NC, USA
²Infinia ML, Inc., Durham, NC, USA. Correspondence to: Dixin Luo <dixin.luo@duke.edu>.

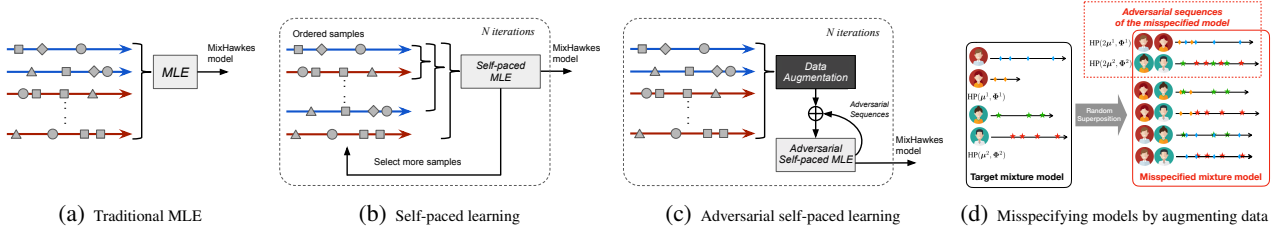


Figure 1. The schemes of various learning methods. The sequences of different Hawkes processes are labeled in different colors.

To overcome the challenges above, we propose a novel adversarial self-paced learning (ASPL) method and train it iteratively to robustly learn mixture models of Hawkes processes. As shown in Figure 1(c), in each iteration we actively generate candidates of “easy” sequences based on data augmentation methods (*e.g.*, random superposition and stitching). Then, based on MLE with an adversarial self-paced (Bengio et al., 2009; Kumar et al., 2010) regularizer, we use these candidate sequences to learn the target model and select “easy” sequences for the next iteration.

The proposed learning method is based on two facts. First, the MixHP model learned from the augmented sequences is always misspecified to some degree, because most of the augmented sequences are not drawn from a Hawkes process. As a result, the augmented sequences still obeying Hawkes processes are adversarial samples of the misspecified model. Second, the easiness of a sample is dependent on the model imposed on it — an easy sample of the target MixHP model can be an adversarial one of the misspecified model. Accordingly, our method selects the adversarial sequences of the current misspecified model to construct the easy sequence set for the target model, with this performed in an iterative manner. With an increase in iterations, the potential easy sequences become dominant in the training set and the misspecified model is revised and approaches to the target one.

2. Adversarial Self-Paced Learning

2.1. Data augmentation and model misspecification

For a single Hawkes process, the over-fitting problem caused by insufficient data can be mitigated based on data augmentation techniques. In particular, the Hawkes process has an interesting superposition property:

Proposition 2.1 ((Xu et al., 2018b)). *Give M independent Hawkes processes with shared impact functions, *i.e.*, $HP(\mu^m, \Phi)$ and $N^m(t) \sim HP(\mu^m, \Phi)$ for $m = 1, \dots, M$, their superposition becomes a single Hawkes process, *i.e.*, $N(t) = \sum_{m=1}^M N^m(t)$ and $N(t) \sim HP(\sum_{m=1}^M \mu^m, \Phi)$.*

Additionally, for a stationary Hawkes process, its impact function $\phi_{cc'}(t)$ satisfies $\int_0^{+\infty} \phi_{cc'}(s) ds < +\infty$, implying that the infectivity of a historical event to current one de-

cays rapidly with respect to the time interval between them, *i.e.*, $\lim_{t \rightarrow +\infty} \phi_{cc'}(t) = 0$. Therefore, given two short sequences belonging to the same Hawkes process, superposing or stitching them can generate a denser or longer sequence for the target Hawkes process model. These two data augmentation strategies have been applied to learn Hawkes processes from imperfect observations (Xu et al., 2018b; 2017; 2018a), which indeed improve learning results.

However, as shown in Figure 1(d), when the sequences generated by different Hawkes processes with different impact functions, their superposition/stitching result will not yield a Hawkes process any more. Therefore, most of the augmented sequences do not obey the target mixture model of Hawkes processes, learning from which leads to a *misspecified MixHP model*, while those really obeying Hawkes processes are the minority of the augmented sequences, will be ignored and treated as *adversarial samples* (Lowd & Meek, 2005; Barreno et al., 2006; Liu & Chawla, 2009; Huang et al., 2011) of the misspecified model.

2.2. The easiness of sequence

Although directly applying traditional data augmentation techniques (*i.e.*, superposing and stitching) is not helpful to learn mixture models of Hawkes processes, the augmented sequences have different levels of fitness with respect to the misspecified model, which provides a reasonable measurement for the easiness of the sequences, and makes self-paced learning possible. Specifically, the likelihood of a sequence under a model reflects the fitness of the model to the sequence. Given a sequence s with I events, we define the easiness of a sequence with respect to the model $\Theta = \{\mu_k, \Phi_k\}_{k=1}^K$:

$$\text{easiness}(s; \Theta) = \max_{k \in \{1, \dots, K\}} \frac{1}{I} \log p(s | \mu^k, \Phi^k). \quad (3)$$

(3) indicates that an easy sample of a mixture model needs to fit one of the clustering component with high probability (even if the probability of the component itself is low). The higher the likelihood is, the easier the sequence is under the given model. Dividing by I , the easiness of the sequences with different lengths becomes comparable. Because (3) is not differentiable, in practice we can use “LogSumExp” operation to achieve a smooth maximum. Accordingly, (3)

can be rewritten as

$$\text{easiness}(\mathbf{s}; \Theta) = \frac{1}{I} \log \left(\sum_{k=1}^K p(\mathbf{s} | \boldsymbol{\mu}^k, \Phi^k) \right). \quad (4)$$

In this case, we define adversarial sequences of our model as those with lowest easiness.

2.3. Proposed learning algorithm

The key idea of our learning method is that the easy sequences of the target mixture model can be the adversarial ones of the current estimated model. When learning a mixture model with potential risk of misspecification based on augmented sequences, we want to find its adversarial sequences and add them into the ‘‘easy’’ sequence set of the target mixture model. The easy sequences are considered in the next training iteration, which are used to revise the misspecified model.

In the m -th learning iteration, given the augmented sequences $\mathcal{S}^{(m)}$ and the easy sequence set $\mathcal{S}_{\text{easy}}$ generated in the previous iteration, we update the current mixture model and select new easy sequences from $\mathcal{S}^{(m)}$ simultaneously, by solving the following max-min optimization problem.

$$\begin{aligned} \max_{\Theta \geq \mathbf{0}} \min_{\mathbf{w} \in \{0,1\}^{|\mathcal{S}^{(m)}|}} & \sum_{\mathbf{s} \in \mathcal{S}^{(m)} \cup \mathcal{S}_{\text{easy}}} \underbrace{\log p(\mathbf{s}; \Theta)}_{\text{log-likelihood}} \\ & + \alpha \sum_{\mathbf{s}_n \in \mathcal{S}^{(m)}} \underbrace{\left[w_n \text{easiness}(\mathbf{s}_n; \Theta) + \zeta(1 - w_n) \right]}_{\text{adversarial self-paced regularizer}}. \end{aligned} \quad (5)$$

Here, $\mathbf{w} = [w_n] \in \{0,1\}^{|\mathcal{S}^{(m)}|}$ is a binary vector, whose element w_n indicates whether \mathbf{s}_n is an easy sequence of the proposed model. The first term represents the log-likelihood of the current model given the whole sequence set, while the second term is the proposed adversarial self-paced regularizer, that measures the easiness of each $\mathbf{s}_n \in \mathcal{S}^{(m)}$ and selects the adversarial sequences of the current model as the easy sequences of the target model. Hyperparameter α controls the significance of the proposed regularizer, and ζ controls the acceptance rate of easy sequences.

We decompose (5) into two sub-problems and solve them via alternating optimization. In each learning iteration, we solve the following two sub-problems:

1) Update current model:

$$\begin{aligned} \hat{\Theta} = \arg \max_{\Theta \geq \mathbf{0}} & \sum_{\mathbf{s} \in \mathcal{S}^{(m)} \cup \mathcal{S}_{\text{easy}}} \log p(\mathbf{s}; \Theta) \\ & + \alpha \sum_{\mathbf{s}_n \in \mathcal{S}^{(m)}} \hat{w}_n \text{easiness}(\mathbf{s}_n; \Theta), \end{aligned} \quad (6)$$

where \hat{w}_n is the indicator learned in the previous step.

2) Select new easy sequences for target model:

$$\begin{aligned} \hat{\mathbf{w}} = \arg \min_{\mathbf{w} \in \{0,1\}^{|\mathcal{S}^{(m)}|}} & \sum_{\mathbf{s}_n \in \mathcal{S}^{(m)}} \left[\zeta(1 - w_n) \right. \\ & \left. + w_n \text{easiness}(\mathbf{s}_n; \hat{\Theta}) \right]. \end{aligned} \quad (7)$$

Algorithm 1 Adversarial Self-Paced Learning for MixHP

- 1: **Input:** Original sequences $\mathcal{S} = \{\mathbf{s}_n\}_{n=1}^N$.
 - 2: **Output:** Parameters $\hat{\Theta}$.
 - 3: Initialize easy sequence set $\mathcal{S}_{\text{easy}} = \emptyset$. Set $m = 0$.
 - 4: **while** $|\mathcal{S}_{\text{easy}}| < 2N$ **do**
 - 5: Augment original sequences and get $\mathcal{S}^{(m)}$.
 - 6: Initialize $\mathbf{w} = \mathbf{0}$
 - 7: **while** Not converge **do**
 - 8: Update $\hat{\Theta}$ via solving (6), and set ζ accordingly.
 - 9: Given ζ , update \mathbf{w} via selecting L sequences $\mathcal{S}_{\text{select}}$ with lowest easiness.
 - 10: **end while**
 - 11: Update easy sequence set: $\mathcal{S}_{\text{easy}} = \mathcal{S}_{\text{easy}} \cup \mathcal{S}_{\text{select}}$.
 - 12: $m = m + 1$.
 - 13: **end while**
-

Maximizing the easiness term in (6) makes the model fit the selected easy sequences and suppresses the influence of those ‘‘non-Hawkes’’ sequences. When selecting new easy sequences, on the contrary, we keep the sequences with low easiness with respect to current model for the following learning iterations. Algorithm 1 shows the scheme of the proposed learning method.

As mentioned in line 8 of Algorithm 1, we set ζ according to the learning result of (6). Given current mixture model $\hat{\Theta}$ and learned distribution of clustering component $\hat{\boldsymbol{\pi}} = [\hat{\pi}_k]$, for all $\mathbf{s}_n \in \mathcal{S}^{(m)}$, we first sort $\text{easiness}(\mathbf{s}_n; \hat{\Theta})$ in ascending order, and then select the top- L augmented sequences. Because the proportion of adversarial sequences in $\mathcal{S}^{(m)}$ can be approximated as $\sum_k \pi_k^2$, the number of easy sequences should not be larger than $\sum_k \pi_k^2 |\mathcal{S}^{(m)}|$. We use $\hat{\pi}_k$ to estimate the real π_k and set $L = \lfloor 0.25 \sum_k \hat{\pi}_k^2 |\mathcal{S}^{(m)}| \rfloor$. Accordingly, $\zeta = \text{easiness}(\mathbf{s}_L; \hat{\Theta})$, where \mathbf{s}_L is the L -th sorted sequence.

2.4. Complexity

Given N sequences with I events per each, the computational complexity for learning a mixture model of K Hawkes processes is $\mathcal{O}(KNI^2)$. Applying the proposed learning strategy, we need to update the model based on various augmented sequence sets in different iterations, and each augmented sequence may have $2I$ events. Denote the maximum number of iterations as M . The computational complexity of our method is $\mathcal{O}(4MKNI^2)$. Fortunately, the proposed learning method is mainly designed for the case of short sequences, whose numbers of events are often very small. Given the improvements on learning results brought from our learning method, which will be shown in the following section, the increase of the computational complexity appears to be tolerable.

Table 1. Comparisons for various methods on real-world data.

| Dataset | Setting | | | MMHP | DMHP | DMHP Stitch | SPL-MixHP | ASPL-MixHP Stitch | ASPL-MixHP Superpose | |
|-----------|----------------------|--------|-----|------|------------|----------------|------------|----------------------|-------------------------|-------------------|
| | N_{train}/N_{test} | T | C | K | loglike | loglike | loglike | loglike | loglike | |
| MIMIC-III | 903 / 226 | 10 yrs | 8 | 10 | -3.46±0.71 | -2.85±0.29 | -2.90±0.20 | -2.66±0.12 | -2.24±0.10 | -2.07±0.08 |
| IPTV | 15103 / 15103 | 24 hrs | 16 | 10 | -0.53±0.13 | 1.38±0.11 | 1.25±0.07 | 1.37±0.09 | 1.45±0.03 | 1.44±0.02 |
| LinkedIn | 1220 / 1219 | 15 yrs | 82 | 5 | -7.39±0.33 | -4.69±0.20 | -4.92±0.14 | -4.64±0.16 | -3.97±0.12 | -4.02±0.14 |

3. Experiments

We denote our adversarial self-paced learning method for mixture models of Hawkes processes as **ASPL-MixHP**. To demonstrate its effectiveness, we compare our method with state-of-the-art methods on three real-world datasets. In particular, we consider four competitive alternatives to our method. *i*) **MMHP**: The multi-task multi-dimensional Hawkes process (Luo et al., 2015), which learns one Hawkes process per sequence and applies K -means to the learned clusters of all sequences. *ii*) **DMHP**: The Dirichlet mixture model of Hawkes processes (Xu & Zha, 2017), which learns the proposed mixture model directly from observed sequences based on variational inference. *iii*) **DMHP-Stitch**: The DMHP model learned based on the augmented sequences generated by random stitching. *iv*) **SPL-MixHP**: The self-paced learning of the mixture model of Hawkes process, which applies the original self-paced learning strategy (Kumar et al., 2010), *i.e.*, in each iteration, we select the sequences with the highest likelihood per event for the next learning iteration, to learn the target mixture model via MLE, as shown in Figure 1(b). For our ASPL-MixHP method, both random superposition and random stitching are applied as feasible data augmentation methods. The hyperparameter α is set to be 10 empirically in the following experiments.

After learning models based on N_{train} training sequences, we evaluate the performance of various methods on N_{test} testing sequences, calculating the average log-likelihood of the testing sequences

$$\text{loglike} = \frac{1}{N_{test}} \sum_{n=1}^{N_{test}} \frac{1}{I_n} p(s_n; \hat{\Theta}). \quad (8)$$

This measurement reflects the fitness of a trained model to the testing samples. Each method is tested in 15 trials. In each trial, the sequences are randomly divided into training and testing sets. The model trained on the training set is applied to the testing set. The averaged testing log-likelihood and its 95% confidence interval are calculated.

We apply our method to *i*) cluster LinkedIn users according to their job-hopping behaviors (Xu et al., 2018a), *ii*) cluster patients according to their admissions (Xu et al., 2017), and *iii*) cluster IPTV users according to their daily viewing records (Luo et al., 2014). These three datasets suffer from data sparsity — generally, each sequence in

these three datasets contains just 10 events or fewer. For the LinkedIn dataset, in each trial the job-hopping behaviors of 1220 LinkedIn users are used to train a mixture model, and the records of the remaining 1219 users are used for testing. These records involve 82 IT companies and universities, which are treated as the event types in the model. For the MIMIC-III dataset, the diseases in patients’ admissions are categorized into 8 classes. For each patient, his/her admissions in ten years are observed event sequences, which are modeled as a mixture model of 10 Hawkes processes. We use 903 sequences for training and 226 sequences for testing in each trial. For the IPTV viewing records, we obtain 15103 daily viewing records of 16 kinds of TV programs from 1000 users in each trial, for training a mixture model of 10 Hawkes processes. The records of the following days are used for testing the model.

Table 1 lists the results of various methods on three real-world datasets. Experimental results show that our ASPL-MixHP method works well, obtaining higher testing log-likelihood than the other methods.

4. Conclusion and Future Work

We propose an adversarial self-paced learning method for mixture models of Hawkes processes. Our method combines data augmentation techniques with a self-paced learning strategy, generating and selecting easy sequences iteratively for the target model, from the adversarial sequences of a potentially misspecified model. We test our method on real-world datasets and demonstrate its potential to improve learning results in cases with short training sequences. In the future, we plan to further reduce its computational complexity and improve its scalability to imbalanced large-scale clustering problems. Beyond mixture models of Hawkes processes, we will extend the proposed method to the mixture models of other temporal point processes.

Acknowledgments This research was supported in part by DARPA, DOE, NIH, ONR and NSF.

References

- Barreno, M., Nelson, B., Sears, R., Joseph, A. D., and Tygar, J. D. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pp. 16–25. ACM, 2006.
- Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48. ACM, 2009.
- Hawkes, A. G. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.
- Huang, L., Joseph, A. D., Nelson, B., Rubinstein, B. I., and Tygar, J. Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, pp. 43–58. ACM, 2011.
- Johnson, A. E., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.
- Kumar, M. P., Packer, B., and Koller, D. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pp. 1189–1197, 2010.
- Liu, W. and Chawla, S. A game theoretical model for adversarial learning. In *2009 IEEE International Conference on Data Mining Workshops*, pp. 25–30. IEEE, 2009.
- Lowd, D. and Meek, C. Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 641–647. ACM, 2005.
- Luo, D., Xu, H., Zha, H., Du, J., Xie, R., Yang, X., and Zhang, W. You are what you watch and when you watch: Inferring household structures from IPTV viewing data. *IEEE Transactions on Broadcasting*, 60(1):61–72, 2014.
- Luo, D., Xu, H., Zhen, Y., Ning, X., Zha, H., Yang, X., and Zhang, W. Multi-task multi-dimensional Hawkes processes for modeling event sequences. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pp. 3685–3691. AAAI Press, 2015.
- Xu, H. and Zha, H. A Dirichlet mixture model of Hawkes processes for event sequence clustering. In *Advances in Neural Information Processing Systems*, pp. 1354–1363, 2017.
- Xu, H., Luo, D., and Zha, H. Learning Hawkes processes from short doubly-censored event sequences. In *International Conference on Machine Learning*, pp. 3831–3840, 2017.
- Xu, H., Carin, L., and Zha, H. Learning registered point processes from idiosyncratic observations. In *International Conference on Machine Learning*, 2018a.
- Xu, H., Luo, D., Chen, X., and Carin, L. Benefits from superposed Hawkes processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 623–631, 2018b.
- Yang, S.-H. and Zha, H. Mixture of mutually exciting processes for viral diffusion. In *International Conference on Machine Learning*, pp. 1–9, 2013.