Improving Maximum Likelihood Estimation of Temporal Point Process via Discriminative and Adversarial Learning

Junchi Yan[1], **Xin Liu**[2], **Liangliang Shi**[2], **Changsheng Li**[*3], **Hongyuan Zha**[2]
[1] Shanghai Jiao Tong University
[2] East China Normal University
[3] University of Electronic Science and Technology of China
yanjunchi@sjtu.edu.cn, xinchrome@gmail.com
851636947@qq.com, lichangsheng@uestc.edu.cn, zha@sei.ecnu.edu.cn

Abstract

Point process is an expressive tool in learning temporal event sequence which is ubiquitous in real-world applications. Traditional predictive models are based on maximum likelihood estimation (MLE). This paper aims to improve MLE by discriminative and adversarial learning. The initial model is learned by MLE explaining the joint distribution of the occurred event history. Then it is refined by devising a gradient based learning procedure with two complementary recipes: i) mean square error (MSE) that directly reflects the prediction accuracy of the model; ii) adversarial classification loss which induces the Wasserstein distance loss. The hope is that the adversarial loss can add sharpness to the smooth effect inherently caused by the MSE loss. The method is generic and compatible with different differentiable parametric forms of the intensity function. Empirical results via a variant of the Hawkes processes demonstrate its effectiveness of our method.

1 Introduction and Related Work

A major line of research has been devoted to modeling event sequences, especially exploring the continuous timestamp information to learn the underlying dynamics, whereby point process has been a powerful and elegant framework. There is rich literature in point process learning under the maximum likelihood estimation (MLE) framework, is aimed to model the joint distribution of events in the sequence. The learned point processes, with their parameters carrying certain implications, can be either used for relational mining [Zhou *et al.*, 2013], or for event prediction by generating the future events [Du *et al.*, 2016]. This paper is aimed to further improve the prediction capability and stability by introducing new learning techniques to the point processes. The approach is based on the general idea of discriminative learning and adversarial learning, which are relatively ignored in the point process learning literature. We briefly introduce the background.

[*]Changsheng Li is correspondence author. The work is supported by NSFC 61602176, 61672231 and NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization U1609220.

Maximum likelihood model While notable progress has been made in generative modelling, in many applications we are still far from producing realistic samples. One of the key open questions is what objective functions one should use to train and evaluate generative models [Theis *et al.*, 2016]. The model likelihood is often considered the most principled training objective and most research in the past decades have been focused on maximum likelihood estimation (MLE) and approximations thereof. Recently we have witnessed several new learning techniques such as those based on adversarial networks [Goodfellow *et al.*, 2014] and kernel moment matching [Dziugaite *et al.*, 2015], and they are not (at least on the surface) related to maximum likelihood. Such a deviation from MLE is often technically motivated by the fact that the exact likelihood is often intractable. Moreover there exist recent methods which further deliberately differ from MLE to avoid undesired behaviour even when MLE is tractable. One such example is the scheduled sampling strategy [Bengio *et al.*, 2015] which obtains empirically improved performance.

Adversarial learning Adversarial learning techniques e.g. generative adversarial networks (GANs) [Goodfellow *et al.*, 2014; Arjovsky *et al.*, 2017] have been recently applied for data generation in different domains. In theory, GANs can be used to model an arbitrarily complex probability distribution, leading to state-of-the-art results on challenging tasks like image super-resolution [Ledig *et al.*, 2017], and video prediction [Mathieu *et al.*, 2015]. In this paper, we are interested in using adversarial learning for point process based event prediction, which is rarely addressed in existing literature [Xiao *et al.*, 2017a], especially for parametric point process.

Temporal point processes Temporal point processes are fundamental mathematical tool to model event sequences in continuous time space. One key component is its conditional intensity function, defined as the probability of observing an event in an infinitesimal window given the history [Aalen *et al.*, 2008]. Over the decades, different parametric forms of the conditional intensity function are specified, including the Hawkes processes [Hawkes, 1971], self-correcting process [Isham and Westcott, 1979] etc. Based on the given parametric form, existing learning based methods are often devised to optimize the objective of MLE with particular algorithms e.g. sampling technique [Ertekin *et al.*, 2015b], majorization-minimization

solver [Lewis and Mohler, 2011] and the ADMM solver [Zhou *et al.*, 2013] etc. Recently there are also emerging works using recurrent neural networks [Du *et al.*, 2016; Xiao *et al.*, 2017b], with the advantage of dismissing the prespecified parametric assumption which is nontrivial and often requires strong domain expert knowledge. As a matter of fact, parametric point processes still have important advantages as prior knowledge can be incorporated (especially in face of insufficient training data), and there is often a clear physical meaning for the parameters [Zhou *et al.*, 2013].

Motivation This paper focuses on enabling gradient descent learning for differentiable parametric point process model. One natural idea is adopting the mean squared error (MSE) as the discriminative loss (though there is technical gap for its use in parametric point process as will be addressed in the paper). Moreover since the ℓ_2 loss inherently assumes the data is drawn from a Gaussian distribution, it tends to lead to smoothed prediction curves that fit poorly on multimodal distributions [Mathieu *et al.*, 2015]. As a complementary recipe, GAN technique (and its derived new loss e.g. Wasserstein distance) is adopted to measure the authenticity of predicted events. The hope is to push the predicted curves from temporally evenly distributed sequences to those more realistic-looking ones. However, the above gradient based discriminative and adversarial learning paradigm cannot be directly applied for many existing models. Popular models like the Hawkes process [Hawkes, 1971], Reinforced Poisson Process [Shen *et al.*, 2014] predict the future sequence via sampling techniques e.g. the thinning algorithm [Ogata, 1981], where the loss cannot be computed in a closed form (at least in the general case) for gradient backpropogation.

Novelty Beyond the MLE generative learning paradigm, we propose a novel approach by discriminative and adversarial learning of differentiable point processes. Our approach is fundamentally based on the observation that by discretizing the counting process, the prediction can be approximated by recursively computing the intensity integral for prediction a closed-form function. Note in the recent conditional GAN based work [Xiao *et al.*, 2018], a network based generator is used for event prediction. While in this paper, the generator refers to an explicit parametric point process model leading to a different learning mechanism.

2 Preliminaries

2.1 Intensity Function of Temporal Point Processes

In temporal point processes, each observed point (i.e. event) t (a non-negative real-valued timestamp) is an outcome of the process, forming a sequence $S = \{t_i | i \in Z_+\}$. Such sequences are inherently different from time series due to their synchronized nature i.e. the timestamp falls in the continuous domain while time series is formed with equal time interval and the fine-grained time information is lost.

A temporal point process can be characterized by the intensity function, and a cascade (i.e. the observed sequence) can be called a realization of the underlying process. Concretely, denote $N(t, t')$ as the number of points during interval (t, t'), $\Lambda(t) = E[N(0, t)]$ as the expected number of points, the *intensity function* is defined as:

$$\lambda(t) = \frac{d}{dt} E[N(0, t)] = \frac{d}{dt} \Lambda(t) \tag{1}$$

The key to characterize a point process is to find appropriate (parametric) form of intensity functions. Often, two components are considered. One is the intrinsic component whose value reflects the inherent property of the sequence. For instance, consider a paper citation prediction task, i.e. predicting the future citation events of a paper, the properties may refer to the publication venue, topic, author affiliation, etc. The other component is the external effect generating from the previous events, which can be called *predecessor-successor dependency*, thus the intensity functions become history-dependent. For example, it is commonly believed that the citation exhibits a Mathew effect [Wang *et al.*, 2013] such that a paper with more citations tend to have even more in (short) future. This effect is found ubiquitous for instance for equipment failures [Ertekin *et al.*, 2015a], crime [Mohler *et al.*, 2011], merger and acquisition [Yan *et al.*, 2016] etc.

As a popular embodiment, the intensity function of a multi-dimensional Hawkes process [Eichler *et al.*, 2017] and the variants [Liu *et al.*, 2017] can be seen as a superposition of background and history effect:

$$\lambda_d^m(t) = \underbrace{\mu_d^m(t)}_{\text{background}} + \underbrace{\sum_{j:t_j<t} \Gamma_d^{mm_j}(t_j) g_d^m(t - t_j)}_{\text{history effect}} \tag{2}$$

where d refers to event taker, t is timestamp, μ is background, $\Gamma_d(t)$ represents the strength that the preceding event affects the successors and g is the affecting kernel function, m demotes the event dimension i.e. event type. It is affected by the preceding events with dimension m_j. A specification of these terms is given in the experiment.

2.2 MLE Learning and Sampling based Prediction

The traditional way of training a temporal point process model is usually based on maximum likelihood estimation (MLE) [Ogata, 1988; Lewis and Mohler, 2011]. Concretely, for D observed sequences $\mathcal{S} = \{S_1, S_2, ..., S_D\}$, and each sequence is represented as $S_d = \{t_j^d\}_{j=1}^{N_d}$, where $t_i^d \leq t_j^d$ if $i < j$. Let $t_0 = 0$, $t_{N_d} = T_d$ as the observation window, the probability density for point t is expressed as [Rubin, 1972]:

$$f_d(t_j | t_1, \ldots, t_{j-1}) = \lambda_d(t_j) \exp\left(- \int_{t_{j-1}}^{t_j} \lambda_d(t) dt \right),$$

Thus the log-likelihood $\mathcal{L}_d$ on the whole sequence is:

$$\log \prod_{j=1}^{N_d} f(t_j | t_1, \ldots, t_{j-1}) = \sum_{j=1}^{N_d} \log \lambda_d(t_j) - \int_0^{T_d} \lambda_d(t) dt \tag{3}$$

As each timestamp t_j is associated with an event type m_j i.e. the so-called dimension, each sequence is denoted as $S_d = \{(t_j^d, m_j^d)\}_{j=1}^{N_d}$. Hence, we have M interdependent intensity functions $\{\lambda_d^m(t)\}_{m=1}^M$ for each type. The resulting overall log-likelihood $\mathcal{L}$ can be written as:

$$\sum_{d=1}^{D} \sum_{m=1}^{M} \mathcal{L}_d^m = \sum_{d=1}^{D} \left(\sum_{j=1}^{N_d} \log \lambda_d^{m_j}(t_j) - \sum_{m=1}^{M} \int_0^{T_d} \lambda_d^m(t) dt \right) \tag{4}$$

Once the parameters are estimated, we can perform in-sample future event prediction via simulation by adopting Ogata's thinning algorithm [Ogata, 1981], which is a counting process based method that generates a sequence point by point. In essence, it is a non-deterministic and sampling process which prevents the use of gradient based analytical learning. Readers are referred to [Ogata, 1981] for more details.

3 Discriminative and Adversarial Learning

Despite the popularity of the MLE models, in many applications one is more interested in predicting *future* events rather than modeling the joint distribution of *past* events. This suggests the potential value for a discriminative paradigm to directly boost the prediction performance. Specifically, the discriminative loss used in the paper is the widely used mean square error (MSE) between the predicted event distribution and ground truth in a specified time window (via bining e.g. by year). However, it is nontrivial to adapt existing prediction methods e.g. Ogata's thinning algorithm [Ogata, 1981]. This paper aims to mitigate this gap.

3.1 Enabling Prediction Error Backpropogation

Maximum likelihood estimation (MLE) learning of point process has achieved empirical success in a number of real-world applications, such as social media popularity dynamics prediction [Shen *et al.*, 2014], citation forecasting [Liu *et al.*, 2017] etc. MLE can obtain a learned generative point process model to fit the observed event sequence. As discussed above, however, in prediction tasks the model is used for future event prediction rather than explaining history. We are aimed to improve the prediction capability via discriminative learning.

In fact, adopting discriminative loss for learning is nontrivial to point process models. The main obstacle lies in the prediction (used to compute the loss against ground truth) is non-deterministic in a forward time window because the prediction in the beginning can affect the prediction later in the window e.g. the Hawkes process like models in Eq. 2. Thus existing methods [Ogata, 1981; Møller and Rasmussen, 2006; 2005; Dassios *et al.*, 2013] dominantly use the traditional counting process based simulation algorithms to estimate the predictions over time, which disallows the error signal of predicted sequence from gradient backpropagating.

For prediction, denote $\widehat{S} = \{\widehat{t}_i\}_{i=1}^{I}$ as the prediction generated via counting process simulation, whose point number in (t_1, t_2) is denoted by $\widehat{N}(t_1, t_2)$. It is desirable to update point process parameters via error propagation by computing the deviation from the actual count $N(t_1, t_2)$. However, $\widehat{N}(t_1, t_2)$ is in general computed by random sampling from the intensity function, which disallows the analytical gradient computing – more details are given as follows.

We propose an approximation method to enable gradient backpropagation. Recall the intensity function in Eq. 2, as well as many other popular forms as listed in Table 1, the main difficulty is that $\lambda_d^m(t)$ is nondeterministic over the prediction window (t_1, t_2). For instance, in the self-exciting Hawkes process, sampling method can generate a few events by some chance during $(t_1, t_1 + \Delta t)$, which can in turn increase the intensity during $(t_1 + \Delta t, t_2)$ (refer to the second term on the right of Eq. 2). Hence, the prediction cannot be exactly computed by the integral $\int_{t=t_1}^{t_2} \lambda_d^m(t)$ in a closed form. Because this will ignore the events that may occur during (t_1, t_2). Similar case also happens for self-correcting point process [Isham and Westcott, 1979] where the recent occurred events decrease the successive event occurrence chance.

Process form	Poisson	Hawkes	Self-correcting	Reactive
History effect	Neutral	Exciting	Inhibiting	Mixtured

Table 1: Popular intensity functions can be decoupled by: $\lambda(t) = \mu(t) + \sum_{t_i<t} \gamma(t, t_i)$, where γ is the temporal kernel quantifying the history event effect to current intensity. Except for Poisson process, in general the prediction over a future time window requires recursively considering the earlier events in that time period, and cannot be computed as a closed-form integer of $\lambda(t)$. This is because the earlier prediction can affect the later predictions, no matter positively (e.g. Hawkes process [Hawkes, 1971]), negatively (e.g. Self-correcting process [Isham and Westcott, 1979]), or both (e.g. Reactive point process [Ertekin *et al.*, 2015a]). Our method is agnostic to these specific parametric forms thus has wide application potential.

Based on the above analysis, our key idea is that for a future time window, one can split it into multiple time units, and the integration is recursively performed on a rolling basis from earlier time units to the later – to get the whole picture, see Eq. 7 where $\widehat{c}_d^m(i)$ is a function of $\{\widehat{c}_d^m(j)\}_{j<i}$. As such, the events occurring in earlier units can be approximately accounted for the prediction in later ones in a more tight fashion.

Formally assume there are D sequences $\mathcal{S} = \{S_1, ..., S_D\}$, with each sequence $S_d = \{(t_j^d, m_j^d)\}_{j=1}^{N_d}$ where m_j^d is the event type, we discretize the continuous timestamp t_j^d by:

$$\overline{t_j^d} = \left\lfloor t_j^d / \tau \right\rfloor \tau = i\tau \tag{5}$$

where τ is the predefined length of the (short) time interval for discretization. The rounding down operation $\lfloor \cdot \rfloor$ transforms a sequence S_d with continuous timestamp within $[0, T_d]$ into a one indexed with discrete time unit (i.e. bining):

$$\overline{S}_d = \{c_d^m(i)\}_{i=0}^{\lfloor T_d/\tau \rfloor}, \quad \text{for} \quad m = 1, 2, \ldots, M \tag{6}$$

where $c_d^m(i) = N(i\tau, i\tau + \tau)$ is the number of events of type m occurring during the $(i+1)$th time interval unit $[i\tau, i\tau + \tau)$ when the time interval is in the *past*. For predicted *future* points, $\widehat{c}_d^m(i) = E[N(i\tau, i\tau + \tau)]$ is the expected event number which in general is a real number.

This transformation enables to generate future points in a differentiable way w.r.t. the point process parameters. More concretely, without loss of generality given the intensity function in Eq. 2, we can write out the analytical approximation formula for each interval $(i\tau, i\tau + \tau)$:

$$\begin{aligned}
\widehat{c}_d^m(i) =& E_{N(i\tau, i\tau+\tau) \sim \mathbb{P}(N(i\tau, i\tau+\tau))}[N(i\tau, i\tau + \tau)] \\
=& \int_{i\tau}^{i\tau+\tau} \Big(\mu_d^m(s) + \sum_{j:t_j<i\tau} \Gamma_d^{mm_j}(t_j) g_d^m(s - t_j) \Big) ds \\
\approx& \sum_{m'=0}^{M} \sum_{j=0}^{i-1} \Gamma_d^{mm'}(j\tau) c_d^{m'}(j) \Big(G_d^m(i\tau + \tau - j\tau) \\
& - G_d^m(i\tau - j\tau) \Big) + U_d^m(i\tau + \tau) - U_d^m(i\tau)
\end{aligned} \tag{7}$$

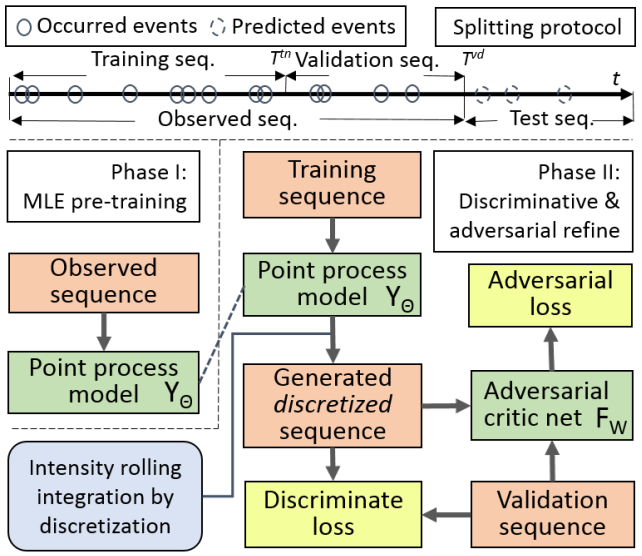


Figure 1: The model is first trained by MLE using the whole observed sequence in line with existing point process learning literature. Then the model is refined by discriminative and adversarial learning. Different from the MLE learning, the refining stage need to split the observed sequence into training and validation one. The latter is used to compute the losses. Note we first use MLE pretraining because the MLE paradigm can utilize the whole observed sequence as training data. Moreover, we find the well-developed MLE technique is more insensitive to initial point compared with our learning approach.

where $\mathbb{P}(\cdot)$ is a distribution determined by the underlying temporal point process, g (and its integration G) is the kernel function, μ (and its integration U) represents intrinsic properties, and Γ represents the predecessor-successor dependency among events. Note the approximate equation in the above formula is a lower-bound (for the Hawkes like self-exciting process) to the true expectation because we only consider the triggering effect from the history citations before $i\tau$ ignoring the possible points occurring during $(i\tau, i\tau+\tau)$. As we assume τ is short such an approximation is reasonable and in practice often becomes natural since the timestamps have limited resolution e.g. year, month in raw data collection.

The above approximation treatment leads to a deterministic and closed-form expectation. We recursively predict future points by Eq. 7 from i-th unit to $(i+1)$th. In each step, $\widehat{c}_d^m$ is a real value regarding with the point process parameters (see Eq. 7). The total event count until time t is written as:

$$\widehat{C}_d(t) = \sum_{m=1}^{M} \sum_{i=0}^{\lfloor \frac{t}{\tau} \rfloor} \widehat{c}_d^m(i) \tag{8}$$

3.2 Discriminative and Adversarial Learning

Figure 1 gives the main flow of our approach. We first apply MLE on the *whole observed sequence* as pre-training to obtain an initial model. This is because we empirically find compared with directly discriminative or adversarial learning, MLE is more insensitive to initial point and can often obtain a good model for further refinement. In fact, there have been well-developed MLE learning techniques [Lewis and Mohler, 2011; Liu *et al.*, 2017] that one can leverage. In contrast, it has been well known that GAN models are practically difficult to train [Arjovsky *et al.*, 2017]. We leave on-the-fly learning of parametric point process model via GAN for future work.

As shown in Fig. 1, we split the whole observed and transformed (see Eq. 6) sequence $\overline{S}_d$ into two segments (recall that we perform in-sample prediction): i) training sequence $\overline{S}_d^{tn} = \{c_d^m(i)\}_{i=0}^{\lfloor T_d^{tn}/\tau \rfloor}$; ii) validation sequence $\overline{S}_d^{vd} = \{c_d^m(i)\}_{i=\lfloor T_d^{tn}/\tau \rfloor+1}^{\lfloor T_d^{vd}/\tau \rfloor}$, i.e. the rest events to validate the correctness of model's prediction $Y_\Theta(\overline{S}_d^{tn})$ conditioned on the input $\overline{S}_d^{tn}$. By this protocol, we can compute discriminative loss e.g. mean squared error (MSE) between these two event count vectors with size $\lfloor T_d^{vd}/\tau \rfloor - \lfloor T_d^{tn}/\tau \rfloor$:

$$\mathcal{L}_{\text{mse}} = E_{\overline{S}_d^{tn} \sim \mathbb{P}(\overline{S}_d^{tn})} \left[\left\| Y_\Theta(\overline{S}_d^{tn}) - \overline{S}_d^{vd} \right\|_2 \right] \tag{9}$$

where $\mathbb{P}(\overline{S}_d^{tn})$ is the distribution of training sequence $\overline{S}_d^{tn}$ and $N_{acc}(\cdot)$ denotes the accumulated total event count in a sequence or period. Given $\overline{S}_d^{tn}$, the model $Y_\Theta(\cdot)$ recursively predicts $\{\widehat{c}_d^m(i)\}_{i=\lfloor T_d^{tn}/\tau \rfloor+1}^{\lfloor T_d^{vd}/\tau \rfloor}$ based on Eq. 7, which establishes the analytical relation between Θ and the MSE loss.

Due to the limitation that the MSE loss inherently assumes the data is drawn from a Gaussian distribution, and works poorly with multimodal distributions [Mathieu *et al.*, 2015], we apply adversarial training technique to provide the multimodality for generated events. The objective function of our GAN model is written as:

$$\mathcal{L}_{\text{gan}} = \begin{cases} \underbrace{E_{\overline{S}_d^{tn} \sim \mathbb{P}(\overline{S}_d^{tn})} \left[F_W \left(Y_\Theta(\overline{S}_d^{tn}) \right) \right] - E_{\overline{S}_d^{vd} \sim \mathbb{P}_r(\overline{S}_d^{vd})} \left[F_W(\overline{S}_d^{vd}) \right]}_{\text{for training critic } F_W} \\ \underbrace{-E_{\overline{S}_d^{tn} \sim \mathbb{P}(\overline{S}_d^{tn})} \left[F_W \left(Y_\Theta(\overline{S}_d^{tn}) \right) \right]}_{\text{for training generator } Y_\Theta} \end{cases} \tag{10}$$

where $\mathbb{P}_r(\cdot)$ is the distribution of observed real sequence $\overline{S}_d$. The critic $F_W(\cdot)$ is a network capable of measuring the Wasserstein distance [Arjovsky *et al.*, 2017] between distribution of generated sequence $\mathbb{P}_g$ and that of real one $\mathbb{P}_r$. $F_W(\cdot)$ is trained to distinguish from real sequences from model generation. The resulting weighted loss is:

$$\mathcal{L}_{\text{mix}} = \delta \mathcal{L}_{\text{gan}} + (1-\delta)\mathcal{L}_{\text{mse}}, \quad \delta \in [0,1] \tag{11}$$

In summary, given a parameterized model $Y_\Theta(\cdot)$ and observed sequences $\mathcal{S}$, first we estimate parameter Θ via MLE on $\mathcal{S}$ to obtain a good baseline model. Then we refine $Y_\Theta(\cdot)$ via the proposed adversarial and discriminative learning. The overall learning method is sketched in Algorithm 1.

Note we use the Wasserstein GAN technique [Arjovsky *et al.*, 2017] (specifically the gradient clipping see line 15 in Algorithm 1) instead of the improved WGAN [Gulrajani *et al.*, 2017]. Because we encounter the persistent gradient vanishing issue regarding with the discriminator's penalty terms accounting for the Lipschitz condition as required in improved WGAN. We get more stable convergence by WGAN.

Algorithm 1 Improving MLE based temporal point process modeling via adversarial and discriminative learning.

Require: Observation $\{S_d^{ob} = [S_d^{tn}, S_d^{vd}]\}_{d=1}^D$ (see Fig. 1); its discretization $\overline{S}_d^{ob}$; event taker d's profile features $\mathbf{x}_d$;
Require: Notations: $Y_\Theta(\cdot)$: generator; $F_W(\cdot)$: adversarial critic network; ξ: the MLE learning early stop threshed; $N_{acc}(\cdot)$: accumulated number of events in a period.
Initialize point process model parameter Θ randomly;
for t in 1 **to** t_{mle} **do**
  Update Θ by EM based estimation for $\mathcal{L}(\Theta)$ by Eq. 4 on the observed sequence $\overline{S}_d^{ob}$;
  Compute prediction error on validation sequence: $r_t = \sum_{d=1}^D |N_{acc}\left(Y_\Theta(\overline{S}_d^{tn})\right) - N_{acc}(\overline{S}_d^{vd})|/N_{acc}(\overline{S}_d^{vd})$
  If $r_{t-1} - r_t < \xi$, break (early stopping);
end for
Initialize critic network parameter W randomly;
while Θ not converge **do**
  Sample $\{\overline{S}_d^{tn}\}_{d=1}^{D_0} \sim \mathbb{P}(\overline{S}_d^{tn})$ a batch of D_0 sequences
  $\Theta \leftarrow \Theta - \eta\nabla_\Theta \sum_{d=1}^{D_0}\Big(-\delta F_W\left(Y_\Theta(\overline{S}_d^{tn})\right) + (1-\delta)\|Y_\Theta(\overline{S}_d^{tn}) - \overline{S}_d^{vd}\|_2\Big)$ by Eq. 11 and Eq. 9, 10;
  for t in 1 **to** t_{post} **do**
    Sample $\{\overline{S}_d^{vd}\}_{d=1}^{D_0} \sim \mathbb{P}_r$ for validation sequences;
    Sample $\{\overline{S}_d^{tn}\}_{d=1}^{D_0} \sim \mathbb{P}$ for training sequences;
    $W \leftarrow W - \eta\nabla_W\left(F_W\left(Y_\Theta(\overline{S}_d^{tn})\right) - F_W(\overline{S}_d^{vd})\right)$
    $W \leftarrow \text{clip}(W, -c, c)$
  end for
end while

3.3 Comparison to Network based Methods

Now we discuss the specific difference of our method to recent work [Xiao *et al.*, 2017a] and its more recent conditional GAN version [Xiao *et al.*, 2018] using network based Wasserstein learning for point process: our model predicts the individual's future events' timestamp and marker (superscript m in Eq. 2) by given its observed preceding sequences. It also means our method allows for personalized model learning for each individual event taker (subscript d in Eq. 2). While in [Xiao *et al.*, 2017a], the model is devised inherently shared for all event takers. Moreover it cannot perform in-sample prediction i.e. for event taker d, predicting its future events based on the history. Instead, the whole sequence can only be generated from scratch in [Xiao *et al.*, 2017a] which hinders the applicability in real-world prediction problems.

More importantly, models in [Xiao *et al.*, 2017a; 2018] are all neural networks making gradient descent trivial. In fact how to adapt to parametric point process calls for more intellectual challenge, which is addressed in this paper. Since parametric point process can have its advantage in terms of better leveraging prior knowledge in face of small data, and higher interpretability compared with black box neural networks, we believe our contribution is orthogonal to [Xiao *et al.*, 2017a; 2018]. In fact, to our best knowledge, we identified no relevant work on adversarial learning of parametric point process.

4 Embodiment and Experiments

4.1 Prediction Performance Evaluation Metrics

Denote $\widehat{C}_d(t) = \widehat{N}_d(0,t)$ as the predicted event number for sequence d before t and $C_d(t)$ as its actual number. Two popular metrics are used to measure the long-term prediction capability in literature [Shen *et al.*, 2014; Liu *et al.*, 2017]:

Mean Absolute Percentage Error (MAPE) It measures the mean *relative* deviation between predicted and true points count. MPAE is given by (the lower the better):

$$MAPE(t) = \frac{1}{D}\sum_{d=1}^{D}\left|(\widehat{C}_d(t) - C_d(t))/C_d(t)\right| \tag{12}$$

Accuracy It measures the fraction of sequences correctly predicted for a predefined error tolerance ϵ. The accuracy of popularity prediction is defined by (the higher the better):

$$ACC(t) = \frac{1}{D}\sum_{d=1}^{D}\left|d : \left|(\widehat{C}_d(t) - C_d(t))/C_d(t)\right| \le \epsilon\right| \tag{13}$$

Note the above definitions are regarding with a time period $[0, t)$ whereby the total event number is accounted for.

Intensity function modeling We use the intensity function model i.e. Eq. 2 as developed in [Liu *et al.*, 2017] for sequence prediction. Here we briefly introduce the components and readers are referred to the papers for details:

$$\mu_d^m(t) = \sum_{p=1}^{P}\beta^m x_d e^{-\theta_d^m t}, \quad g_d^m(t) = e^{-\omega_d t}, \quad \Gamma_d^{mm'}(t) = \alpha_d^{mm'} \tag{14}$$

where model parameter Θ includes α, β, θ and ω, except for $x_d \in \mathbb{R}^P$ denoting properties associated with event performer e.g. a paper with its publication venue, institution and its citation events. Note we omit the index p in summation of all profile features for β_m, x_d, θ_d^m. By Eq. 4 and Eq. 2, the overall MLE log-likelihood function is:

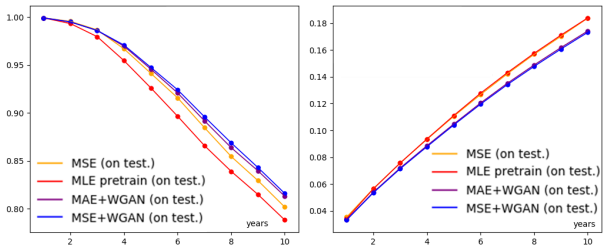
$$\mathcal{L}(\Theta) = \sum_{d=1}^{D}\left(\sum_{j=1}^{N_d}\log\lambda_d^{m_j}(t_j) - \sum_{m=1}^{M}U_d^m(T_d) - \sum_{m=1}^{M}\sum_{j=1}^{N_d}\Gamma_d^{mm_j}(t_j)G_d^m(T_d - t_j)\right) + r(\Theta) \tag{15}$$

where $r(\Theta)$ is an optional sparsity regularizer. An EM algorithm can be used for MLE based parameter estimation. Details can be found in [Liu *et al.*, 2017].

4.2 Dataset and Settings

Synthetic data We simulate $D = 1000$ synthetic sequences by using the above intensity function over $[0, T)$ for $T = 25$.

The observation window spans $[0, 15]$, consisting of $[0, 10]$ as training window and $[10, 15]$ as validation. The testing period is $[15, 25]$. For model's parameters in Eq. 2 and Eq. 14, event take d's profile feature is of size $P = 15$ i.e. $\beta^m, x_d \in \mathbb{R}^{15}$. For each event taker, its profile value x_d is sampled by a uniform distribution $\mathcal{U}(0, 1)$. The profile encoding parameter β^m is sampled from $\mathcal{U}(0, 3)$, and θ_d^m is shared among all takers d and is sampled from $\mathcal{U}(0, 0.5)$. The triggering parameter α_d is set varying over takers and has the form of $\alpha_d = c *$



(a) Accuracy on citation data. (b) MAPE on citation data.

Figure 2: Prediction performance evaluation ($\epsilon = 0.3$ for ACC) over years on the real-world citation dataset [Sinha *et al.*, 2015]. Mean absolute error (MAE) is also tested by replacing the MSE loss.

$\exp(-0.01\sqrt{d})$ where c is a sampled from $\mathcal{U}(0, 6)$ and d is the d-th sequence from 1 to D. Given the above parameters, the event sequence can be readily simulated using off-the-shelf algorithm e.g. the thinning method [Ogata, 1981].

To study the robustness of our model, we further add noise to the timestamp of each event. Not for learning, we use the noise-add data, while for performance evaluation of the test period, we compare with the ground truth without noise.

Real-world dataset Microsoft Academic Graph (MAG) [Sinha *et al.*, 2015] consists scientific publications with their individual profiles e.g. authors, institutions, venues etc. and citation records over time. We have collected 10,000 publications satisfying: i) published in computer science venues during 1980-1990; ii) each paper has at least 5 citations. For each of these papers, we collect a span of 25 years of citation records since its publishing year. Following the protocol in Fig. 1, the 25-year is split into the observed period (from year 1 to 15) and the test period (from year 16 to 25). Moreover, the observed period is divided into the training period (from year 1 to 10) and the validation period (from year 11 to 15).

We also use NYSE transaction [Du *et al.*, 2016] containing 0.7 million high-frequency transaction records at a stock one day, forming 3200 sequences. We pick the top 1000 sequences with most events for our experiments.

Protocols We use convolutional neural network (CNN) [LeCun *et al.*, 1998] as the critic for classifying genuine sequence $\overline{S}_d^{vd}$ from the generated prediction $Y_\Theta(\overline{S}_d^{tn})$. The CNN has a conv-pooling-conv-FC structure with 128 filters. For WGAN [Arjovsky *et al.*, 2017], we set parameter clip $c = 0.1$, learning stepsize $\eta = 0.01$, mixed loss weight $\delta = 0.5$; For detecting overfitting and early stopping, we set $\xi = 0.001$. RMSProp [Tieleman and Hinton, 2012] is used for backpropagation. For evaluation metrics, we use MAPE (Eq. 12) and ACC (Eq. 13) with different tolerance ϵ in line with [Shen *et al.*, 2014; Liu *et al.*, 2017]. We test MSE (mean square error) as discriminative loss. Three settings are evaluated: i) MLE alone, ii) MLE_{MSE}, iii) $\text{MLE}_{\text{MSE+WGAN}}$ where the subscripts denote the posterior learning after MLE.

4.3 Results and Discussion

Synthetic data results From the results as shown in Table 2 one can observe our technique can improve MLE.

Real-world data results Table 3 reports both the MAPE and ACC over the first 5 years and on the whole 10 years (including the first 5-year) respectively. The results show the posterior learning, especially with the mixed loss involving both discriminative and adversarial learning can often improve the MLE baseline. Fig. 2 discloses the yearly performance over the 10-year testing period. One can see combing MSE with WGAN outpeforms MSE only, which suggests of adopting adversarial loss in addition with MSE. Similar results can be found in Table 3 for the NYSE transaction dataset. Moreover, we study the behavior of the weight δ between MSE and WGAN loss in Eq. 11 as shown in Table 4.

	method	MAPE	$\text{ACC}_{\epsilon=.3}$	$\text{ACC}_{\epsilon=.2}$	$\text{ACC}_{\epsilon=.1}$
5 yrs.	MLE	7.46	99.9	99.0	69.8
	MLE_{MSE}	7.13	100.0	99.6	72.8
	MLE_{WGAN}	5.95	100.0	99.6	83.0
	$\text{MLE}_{\text{MSE+WGAN}}$	**5.55**	**100.0**	**99.4**	**85.8**
10 yrs.	MLE	11.87	98.8	84.8	43.0
	MLE_{MSE}	11.13	99.1	87.2	47.8
	MLE_{WGAN}	8.98	99.3	93.2	63.4
	$\text{MLE}_{\text{MSE+WGAN}}$	**8.72**	**99.6**	**94.6**	**65.9**

Table 2: Results ($\delta = 0.3$) on 1000 sequences via simulation.

	method	MAPE	$\text{ACC}_{\epsilon=.3}$	$\text{ACC}_{\epsilon=.2}$	$\text{ACC}_{\epsilon=.1}$
5 yrs.	MLE	6.46	100.0	98.80	86.76
	MLE_{MSE}	4.21	100.0	100.0	95.75
	MLE_{WGAN}	4.10	100.0	100.0	96.23
	$\text{MLE}_{\text{MSE+WGAN}}$	**3.87**	**100.0**	**100.0**	**96.63**
10 yrs.	MLE	11.98	95.59	87.25	51.64
	MLE_{MSE}	7.42	99.60	97.03	79.87
	MLE_{WGAN}	7.19	99.84	97.43	81.40
	$\text{MLE}_{\text{MSE+WGAN}}$	**6.76**	**99.68**	**97.67**	**83.48**
5 yrs.	MLE	10.85	93.78	84.07	59.32
	MLE_{MSE}	11.11	92.63	82.46	59.23
	$\text{MLE}_{\text{MSE+WGAN}}$	10.48	94.59	85.07	62.25
	$\text{MLE}_{\text{MAE+WGAN}}$	**10.41**	**94.77**	**85.66**	**62.82**
10 yrs.	MLE	18.38	79.57	64.81	39.75
	MLE_{MSE}	18.41	78.84	63.96	40.87
	$\text{MLE}_{\text{MSE+WGAN}}$	17.43	81.32	66.64	42.77
	$\text{MLE}_{\text{MAE+WGAN}}$	**17.33**	**81.61**	**66.99**	**42.82**

Table 3: Results ($\delta = 0.5$) on real-world data from NYSE transaction (top half) and Microsoft Academic Graph (bottom half). Using WGAN alone in our experiments on citation data failed to converge.

	weight	MAPE	$\text{ACC}_{\epsilon=.3}$	$\text{ACC}_{\epsilon=.2}$	$\text{ACC}_{\epsilon=.1}$
5 years	$\delta = 0.1$	5.75	100.0	99.4	85.2
	$\delta = 0.3$	**5.55**	**100.0**	**99.4**	**85.8**
	$\delta = 0.5$	5.83	100.0	99.6	83.2
	$\delta = 0.7$	5.90	100.0	99.6	83.4
	$\delta = 0.9$	5.93	100.0	99.6	83.4
10 years	$\delta = 0.1$	8.77	99.2	94.4	64.4
	$\delta = 0.3$	**8.72**	**99.6**	**94.6**	**65.9**
	$\delta = 0.5$	8.82	99.6	94.1	65.2
	$\delta = 0.7$	8.95	99.6	93.4	63.9
	$\delta = 0.9$	8.97	99.6	93.0	63.4

Table 4: Weight δ sensitivity in Eq. 11 on 1000 simulated sequences.

5 Conclusion

This paper presents a novel technique for improving maximum likelihood based estimation of temporal point processes. Its utility is verified by testing multi-dimensional Hawkes-like point process on synthetic and real-world sequences. Our model is parametric in contrast to those full RNNs models.

For synergizing the MSE and WGAN losses, in this paper we only tried a naive weighted linear sum with moderate improvement to each other. Seeing the practical complexity of training GAN, better tricks e.g. gradually adding GAN loss may be developed which we leave for future work.

References

[Aalen *et al.*, 2008] O. Aalen, O. Borgan, and H. Gjessing. *Survival and event history analysis: a process point of view*. Springer Science & Business Media, 2008.

[Arjovsky *et al.*, 2017] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *ICML*, 2017.

[Bengio *et al.*, 2015] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *NIPS*, 2015.

[Dassios *et al.*, 2013] A. Dassios, H. Zhao, et al. Exact simulation of hawkes process with exponentially decaying intensity. *Electronic Communications in Probability*, 18(62):1–13, 2013.

[Du *et al.*, 2016] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song. Recurrent marked temporal point processes: Embedding event history to vector. In *KDD*, 2016.

[Dziugaite *et al.*, 2015] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *UAI*, 2015.

[Eichler *et al.*, 2017] M. Eichler, R. Dahlhaus, and J. Dueck. Graphical modeling for multivariate hawkes processes with nonparametric link functions. *Journal of Time Series Analysis*, 2017.

[Ertekin *et al.*, 2015a] S. Ertekin, C. Rudin, and T. McCormick. Reactive point processes: A new approach to predicting power failures in underground electrical systems. *The Annals of Applied Statistics*, 9(1):122–144, 2015.

[Ertekin *et al.*, 2015b] S. Ertekin, C. Rudin, and T. H McCormick. Reactive point processes: A new approach to predicting power failures in underground electrical systems. *The Annals of Applied Statistics*, 2015.

[Goodfellow *et al.*, 2014] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[Gulrajani *et al.*, 2017] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein gans. In *NIPS*, 2017.

[Hawkes, 1971] A. Hawkes. Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 438–443, 1971.

[Isham and Westcott, 1979] V. Isham and M. Westcott. A self-correcting point process. *Stochastic Processes and Their Applications*, 8(3):335–347, 1979.

[LeCun *et al.*, 1998] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Ledig *et al.*, 2017] C. Ledig, L. Theis, F. Huszar, J. Caballero, and A. Cunningham. Photo realistic single image super resolution using a generative adversarial network. In *CVPR*, 2017.

[Lewis and Mohler, 2011] E. Lewis and E. Mohler. A nonparametric em algorithm for multiscale hawkes processes. *Journal of Nonparametric Statistics*, 2011.

[Liu *et al.*, 2017] X. Liu, J. Yan, S. Xiao, X. Wang, H. Zha, and S. Chu. On predictive patent valuation: Forecasting patent citations and their types. In *AAAI*, 2017.

[Mathieu *et al.*, 2015] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2015.

[Mohler *et al.*, 2011] G. Mohler, M. Short, J. Brantingham, F. Schoenberg, and G. Tita. Self-exciting point process modeling of crime. *Journal of the American Statistical Association*, 2011.

[Møller and Rasmussen, 2005] J. Møller and J. G Rasmussen. Perfect simulation of hawkes processes. *Advances in applied probability*, 37(03):629–646, 2005.

[Møller and Rasmussen, 2006] J. Møller and J. G Rasmussen. Approximate simulation of hawkes processes. *Methodology and Computing in Applied Probability*, 8(1):53–64, 2006.

[Ogata, 1981] Y. Ogata. On lewis' simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1):23–31, 1981.

[Ogata, 1988] Y. Ogata. Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical Association*, 1988.

[Rubin, 1972] I. Rubin. Regular point processes and their detection. *IEEE Transactions on Information Theory*, 18(5):547–557, 1972.

[Shen *et al.*, 2014] H. Shen, D. Wang, C. Song, and A. Barabasi. Modeling and predicting popularity dynamics via reinforced poisson processes. In *AAAI*, 2014.

[Sinha *et al.*, 2015] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B. Hsu, and K. Wang. An overview of microsoft academic service (mas) and applications. In *WWW*, pages 243–246, 2015.

[Theis *et al.*, 2016] L. Theis, A. van den Oord, and M. Bethge. A note on the evaluation of generative models. In *ICLR*, 2016.

[Tieleman and Hinton, 2012] T. Tieleman and G. Hinton. Rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 2012.

[Wang *et al.*, 2013] D. Wang, C. Song, and A. Barabási. Quantifying long-term scientific impact. *Science*, 342(6154):127–132, 2013.

[Xiao *et al.*, 2017a] S. Xiao, M. Farajtabar, X. Ye, J. Yan, L. Song, and H. Zha. Wasserstein learning of deep generative point process models. In *NIPS*, 2017.

[Xiao *et al.*, 2017b] S. Xiao, J. Yan, X. Yang, H. Zha, and S. Chu. Modeling the intensity function of point process via recurrent neural networks. In *AAAI*, 2017.

[Xiao *et al.*, 2018] S. Xiao, J. Yan, C. Li, B. Jin, X. Wang, X. Yang, S. Chu, and H. Zha. Learning conditional generative models for temporal point processes. In *AAAI*, 2018.

[Yan *et al.*, 2016] J. Yan, S. Xiao, C. Li, B. Jin, X. Wang, B. Ke, X. Yang, and H. Zha. Modeling contagious merger and acquisition via point processes with a profile regression prior. In *IJCAI*, 2016.

[Zhou *et al.*, 2013] K. Zhou, H. Zha, and L. Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *AISTATS*, 2013.