The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Learning Conditional Generative Models for Temporal Point Processes

Shuai Xiao,[†] Honteng Xu,[‡] Junchi Yan,[†§*] Mehrdad Farajtabar,[◇]
Xiaokang Yang,[†] Le Song,[◇] Hongyuan Zha[◇]
[†] Shanghai Jiao Tong University
[◇]College of Computing, Georgia Institute of Technology
[‡] Duke University, [§] IBM Research – China
{benjaminforever,yanjunchi,xkyang}@sjtu.edu.cn
hongteng.xu@duke.edu, mehrdad@gatech.edu
{lsong,zha}@cc.gatech.edu

Abstract

Estimating the future event sequence conditioned on current observations is a long-standing and challenging task in temporal analysis. On one hand for many real-world problems the underlying dynamics can be very complex and often unknown. This renders the traditional parametric point process models often fail to fit the data for their limited capacity. On the other hand, long-term prediction suffers from the problem of bias exposure where the error accumulates and propagates to future prediction. Our new model builds upon the sequence to sequence (seq2seq) prediction network. Compared with parametric point process models, its modeling capacity is higher and has better flexibility for fitting real-world data. The main novelty of the paper is to mitigate the second challenge by introducing the likelihood-free loss based on Wasserstein distance between point processes, besides negative maximum likelihood loss used in the traditional seq2seq model. Wasserstein distance, unlike KL divergence i.e. MLE loss, is sensitive to the underlying geometry between samples and can robustly enforce close geometry structure between them. This technique is proven able to improve the vanilla seq2seq model by a notable margin on various tasks.

Introduction

The ability of looking into the future is a challenging but luring task. People are willing to estimate the occurrence probability for their interested events so that they can take preemptive action. For example, after reviewing the admission history of patients, the doctors may give kind warning for the patients who are at high risk of certain diseases. When having access to working experience of job seekers, headhunters can evaluate one's future career path and recommend a suitable position at proper time. In these cases, the historical observations always provide us with important guidance to predict future events — not only the order of events but also the time span between them contain useful information about the underlying dynamics of the process. Therefore, we need to predict the distribution of future events conditioned on the partial historical observations and the estimated dynamics of the process.

However, this task is very challenging mainly for two reasons: First, we have little knowledge on what model the whole process actually follows. For real-world events, the underlying generating process is often unknown. If a misspecified model is used to fit the observed data, its out-of-sample generality would degrade. Therefore, a flexible model with enough capacity to handle such a complexity is necessary. Second, long-term prediction suffers from the so-called bias exposure problem (Bengio et al. 2015) where the error accumulates and propagates to latter prediction. Most work relies on the one-step maximum likelihood estimation. As a result, the high-order dependency over events for likelihood function has not been fully explored.

In this paper, we propose a new neural network-based sequential model and design a novel learning algorithm to address the above two challenges, respectively. In particular, our model is a kind of neural point process models (Du et al. 2016; Xiao et al. 2017c), which is very robust to the problem of model misspecification and has large capacity. In the learning phase of our model, we design a new algorithmic framework combining traditional maximum likelihood estimation (MLE) with Wasserstein distance-based learning. Our learning method is based on the following two facts: On one hand, MLE loss or KL divergence requires strict matching between two probability distributions and is non-biased estimation of parameters, which is sensitive to sample noise and outliers; on the other hand, unlike MLE loss, which does not consider how close two samples are but only their relatively probability, Wasserstein distance is sensitive to the underlying geometry structure of samples but has biased gradients(Bellemare et al. 2017). To take advantage of the strengths of these two methods and mitigate the bias exposure in long-term prediction, our method incorporate Wasserstein distance besides MLE — both the KL divergence and the Wasserstein distance between generated and real samples are minimized jointly.

In a nut shell, the main contributions of the paper are:

1. We develop a novel long-term time dependent event sequence prediction model by exploring the Wasserstein loss besides the traditional MLE loss. Differing from the very recent adversarial learning based point process model (Xiao et al. 2017a), which can only estimate the

*Corresponding Author

overall intensity of a set of sequences, our model allows for *individual level* in-sample forward prediction of event sequence conditioned on its history. In particular, we pursuit long-term prediction via a sequence to sequence model and boost its performance via adversarial learning technique. To the best of our knowledge, this is the first temporal event model for individual sequence prediction based on adversarial learning.

2. Our model shows promising performance on various synthetic data and real-world event sequences in different domains. On one hand, our model can outperform state-of-the-art neural network based event prediction methods in terms of prediction error. On the other hand, our generator can fit well with various specified parametric point process models and generates plausible event sequences, which may bear high potential for other applications involving domain-specific data synthesis, especially for the cases that are sensitive and hard/expensive to collect data e.g. medical records (Choi et al. 2017). In other words, our predictive model provides a new framework to synthesize event sequences from history observations.

Related Work

We perform a brief literature review focusing on a few aspects: i) temporal point processes for their dominant usages in event sequence learning; ii) related work about generative adversarial networks as a similar framework used in this paper; iii) miscellaneous event prediction approaches especially those based on recurrent neural networks which are also the building blocks of our approach.

Temporal point processes

Point processes have been widely used to model and predict time-dependent event sequences, e.g. earthquake and its aftershocks (Ogata 1988), finance transaction analysis (Bacry, Mastromatteo, and Muzy 2015), medical health prediction (Xu et al. 2017). Readers are referred to (Aalen, Borgan, and Gjessing 2008) for a more comprehensive reading for related concepts. Most previous models largely are based maximum likelihood estimation, including the traditional parametric models that specify the parametric form of the underlying conditional intensity function (Rubin 1972; Lewis and Mohler 2011), as well as more recent literature on recurrent network-based intensity function modeling (Du et al. 2016; Xiao et al. 2017c; 2017b), in the hope of increasing the modeling capability. In this paper we work with the temporal event sequences using sequence to sequence networks, instead of explicitly modeling the intensity function from the point process perspective, as our ultimate goal is for long-term prediction. However, our model can be used to generate event sequences fitting with different point processes.

Generative adversarial networks (GANs)

GANs have been extensively studied since the seminal work (Goodfellow et al. 2014), which has proven to be a promising alternative to traditional maximum likelihood-based models (Theis, van den Oord, and Bethge 2015). Its learning procedure involves a minimax game between a generative model and a discriminative model. The vanilla GANs paradigm in (Goodfellow et al. 2014) accepts a random input signal to generate output data, aiming at fitting a certain distribution. Recent studies (Arjovsky, Chintala, and Bottou 2017; Gulrajani et al. 2017) further present improved and more stable learning techniques for GANs, which explain GANs from the viewpoint of Wasserstein distance-based learning strategy.

For sequential data, many approaches have resort to GANs techniques for sequence generation (Press et al. 2017; Zhang et al. 2017). For instance, text generation can be performed without specified input and the generation performance largely relies on expert judgment and vague statistics like n-gram based blue-score. The methods in (Yu et al. 2017) use reinforcement learning algorithm as unbiased gradients for discrete sequence outputs. However, they suffer from high variance of estimated gradients and high-cost computation in practice. Note that all the methods above do not deal with time stamps and only consider the order of events in their learning phases. In contrast, learning the time dependency over events is the main focus of this paper.

Our work is also related to the so-called conditional GAN techniques, whereby the model output is conditioned on the input. Examples include image super-resolution (Ledig et al. 2017), image to image translation (Isola et al. 2016). In general, it performs relatively well in image to image translation with low stochasticity of outputs conditioning on inputs (Isola et al. 2016). Our approach can be regarded as a conditional GANs model, whereby the output, i.e. predicted event sequence, is conditioned on the input observed history sequence. More specifically, we regard our model involves conditional Wasserstein loss (Arjovsky, Chintala, and Bottou 2017; Gulrajani et al. 2017) for learning.

One closely related work is the Wasserstein learning of point processes in (Xiao et al. 2017a). However, because that method aims to train a generator that can generate a few event sequences fitting into the learned distribution, the pipeline of that method is not conditional, i.e., its generator cannot accept specified input to generate expected output. In contrast, our generator need a partial observed sequence and generate its future conditioned on the input history. As a result, our approach is able to perform in-sample prediction instead of mining the distribution for the training data as a whole – which is individual sample agnostic.

Miscellaneous event sequence prediction models

Event sequence prediction is an active research topic in literature, and there have been a large amount of relevant methods based on different techniques. Among them, recurrent neural networks (RNNs) have been recently applied to sequential modeling (Du et al. 2016; Bengio et al. 2015) and achieved state-of-the-art performance. The former (Du et al. 2016) employs one-step maximum likelihood estimation and the later (Bengio et al. 2015) employs a curriculum learning strategy in the training stage to mitigate the error propagation problem. The authors in (Lian et al. 2015) propose a multi-task approach to deal with data sparsity, which has high computational cost. Works use data-driven ap-

proach to predict future activities given auxiliary data from distributed sensors (Minor, Doppa, and Cook 2015), and embedding time to robust features to event prediction (Li 2017). All these methods do not involve adversarial learning, which makes them inherently different from our approach. Meanwhile, the presented technique in this paper is general.

Proposed Framework

Given an observed time-dependent event sequence $\zeta_n = t_1, t_2, \ldots, t_n$, the task is to learn a generative model that follows the optimal distribution $P(t_{n+1}, \ldots, t_{n+m}|\zeta_n)$. To achieve this aim, we design a discriminative learning algorithm based on Wasseratein loss combined with maximum likelihood estimation.

Generative model $g_\theta(\cdot)$

Our forward prediction model follows the sequence-to-sequence approach (Sutskever, Vinyals, and Le 2014) as depicted in Figure 1. The generator firstly encodes the partial observation into compact representation. Then, the decoder recurrently outputs decoded sequence. If the input and output sequences are $\zeta = \{t_1, \ldots, t_n\}$ and $\rho = \{t_{n+1}, \ldots, t_{n+m}\}$, the seq2seq model learns a mapping $g_\theta(\zeta) = \rho$, which maximizes the distribution of real data.

The RNN recurrently embeds the inputs to an intermediate fixed-dimensional representation $\mathbf{h}_i$ incorporating the history information before time t_i, and then maps to the prediction t_{i+1}, according to the following equations:

$$\mathbf{h}_i = \phi_g^h(\mathbf{A}_g^{\mathbf{h}} t_i + \mathbf{B}_g^h \mathbf{h}_{i-1} + \mathbf{b}_g^h), \quad (1)$$

$$t_{i+1} = \phi_g^x(\mathbf{B}_g^x \mathbf{h}_i + b_g^x), \quad (2)$$

where ϕ_g^h, ϕ_g^x are activation functions. In practice, our RNN is an LSTM, which can capture long range temporal dependencies and we follow the same structure as in (Sutskever, Vinyals, and Le 2014).

The RNN estimates the probability $P(\rho|\zeta)$ by recursively computing the conditional probability according to

$$P(\rho|\zeta) = \prod_{i=n}^{n+m-1} P(t_{i+1}|h_i, t_1, \ldots, t_i). \quad (3)$$

The RNN iteratively generates the outputs by selecting the value that maximizes the conditional probability $P(t_{i+1}|h_i, t_1, \ldots, t_i)$.

By maximizing $P(\rho|\zeta)$ given training pairs $< \zeta, \rho >$ of real data, the model parameters can be learned via maximum likelihood estimation. In the learning stage, true data ρ are feed to the RNN to compute the probability while in the inferring stage, previously generated steps are taken as input to RNN. The discrepancy between learning and inferring process can propagate errors to the subsequent generated sequence as conjectured and analyzed in (Bengio et al. 2015). To mitigate this problem, we propose to add Wasseratein loss to the original problem. This brings a major advantage that the *whole* generated sequences can be taken into consideration for learning in the spirit of minimizing the distance between the distribution of generated sequences and real data. We believe this can be complementary to the MLE loss.

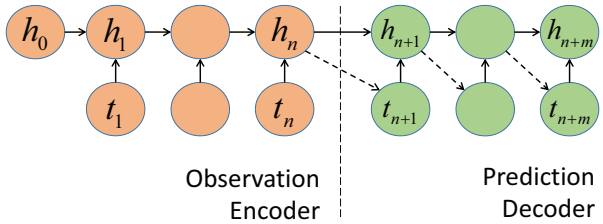


Figure 1: Our generator follows the sequence to sequence recurrent network in line with (Sutskever, Vinyals, and Le 2014). Different from the traditional MLE based learning loss, we update the seq2seq LSTM via a fused loss including both MLE and Wasserstein losses.

Motivated by the above observation, we devise Wasseratein loss-based conditional GAN techniques and aim to learn a joint loss as described in the subsequent subsections.

Wasserstein loss and discriminative model $f_w(\cdot)$

Let $\mathcal{X}$ be a compact metric set where each element $\xi \in \mathcal{X}$ is one realization of point processes and $\mathbb{P}(\mathcal{X})$ denotes the space of probability measures defined on $\mathcal{X}$. Now we can define the divergence between two point process distribution $\mathbb{P}_r, \mathbb{P}_\theta \in \mathbb{P}(\mathcal{X})$. The Wassestein distance between two distributions $\mathbb{P}_r$ and $\mathbb{P}_\theta$ can be defined as:

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_\theta)} \mathbb{E}_{(\xi,\eta)\sim\gamma}[c(\xi, \eta)]$$
$$= \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_\theta)} \int_{\mathcal{X}\times\mathcal{X}} c(\xi, \eta) d\gamma(\xi, \eta), \quad (4)$$

where $\Pi(\mathbb{P}_r, \mathbb{P}_\theta)$ denotes the set of all joint distributions $\gamma(\xi, \eta)$ whose marginals are respectively $\mathbb{P}_r$ and $\mathbb{P}_\theta$, and $c(\xi, \eta)$ is the cost function c: $\mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}^+$.

Solving Equation 4 directly is intractable, thus we turn to its Kantorovich-Rubinstein duality in accordance with (Arjovsky, Chintala, and Bottou 2017):

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{||f||_L \leq 1} \mathbb{E}_{\xi\sim\mathbb{P}_r}[f(\xi)] - \mathbb{E}_{\eta\sim\mathbb{P}_\theta}[f(\eta)], \quad (5)$$

where the supremum is over all the 1-Lipschitz functions $f : \mathcal{X} \mapsto \mathbb{R}$.

For our conditional generator that generates $\hat{\rho} = g_\theta(\zeta)$ based on ζ, the Wasseratein distance can be rewritten as:

$$W(\mathbb{P}_r, \mathbb{P}_\theta) =$$
$$\sup_{||f||_L \leq 1} \mathbb{E}_{\{\zeta,\rho\}\sim\mathbb{P}_r}[f(\{\zeta, \rho\})] - \mathbb{E}_{\zeta\sim\mathbb{P}_r}[f(\{\zeta, g_\theta(\zeta)\})], \quad (6)$$

where $\{\zeta, \rho\}$ represents the long sequence generated by stitching the observation ζ with the prediction ρ.

The key part of Wasserstein learning is the design of the 1-Lipschitz function f. In theory, the function space of 1-Lipschitz function is very large. In practice, it's hard to fully explore the whole space of 1-Lipschitz function but we can apply a neural network to approximate the function and learn it under the 1-Lipschitz constraint. In this work, we present

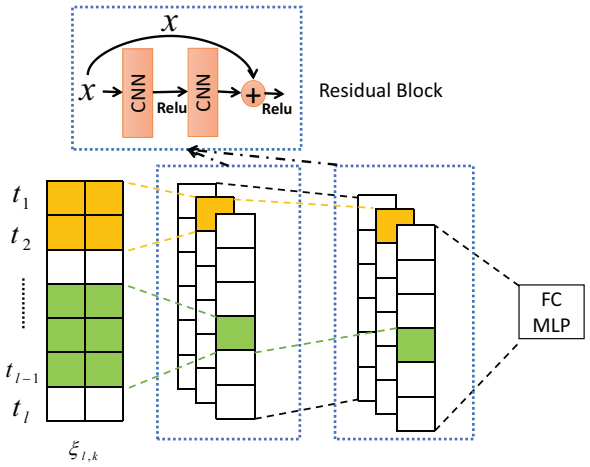


Figure 2: Our discriminator's structure follows a one-dimensional CNN architecture in line with (Gulrajani et al. 2017). The input is l length sequence, where each element has k features. In the figure $k = 2$, while in our paper, $k = 1$, which is the inter-event time duration.

the design and implementation for the discriminator module $f_w(\cdot)$ with parameters w. Specifically, we choose the Residual convolutional neural network (CNN) (He et al. 2016) shown in Figure 2 to approximate the 1-Lipschitz function $f_w(\cdot)$, which proves to work efficiently (Gulrajani et al. 2017). The parametric approximation of function $f_w(\cdot)$ with direct connection is highly flexible and has much power to approximate a large scope of functions. The network consists of two layered residual blocks, where each block is CNN-based residual network, and the final layer is a fully connected forward network.

Final objective for minmax learning

Previous work (Isola et al. 2016; Gulrajani et al. 2017) shows that in general it may be beneficial to mix the maximum likelihood loss and adversarial loss (specifically Wasserstein loss $W(\cdot,\cdot)$ in this paper). In the context of sequence learning, we believe the benefit is more direct and perceivable as the MLE is based on recursive prediction from one step to the next, while the Wasserstein loss can measure the deviation between two sets of sequences as a whole. Therefore, the combined loss can be written by:

$$W(\mathbb{P}_r, \mathbb{P}_\theta) - \sigma \log(P_\theta(\rho|\zeta)) \tag{7}$$

where σ is the trade-off between Wasserstein loss and negative log-likelihood.

In summary, our objective function is a minmax problem whose objectives can be optimized alternatively. Following the work in (Gulrajani et al. 2017), the 1-Lipschitz constraint of function f_w can be converted as a regularization term to the Wasserstein loss, leading to the final loss function regarding with the generator's parameter θ and discriminator's

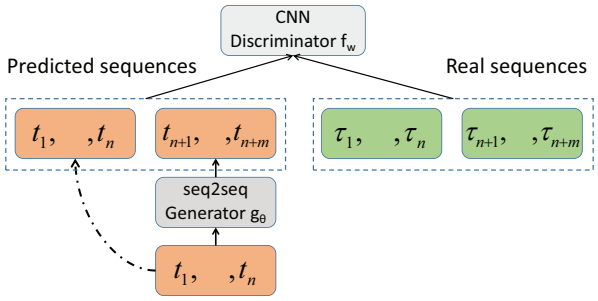


Figure 3: The overall framework of conditional generative model. The discriminator (embodied by CNN in the paper) computes real-valued scores for both synthetic sequences by the generator and real sequences, which are used to compute the Wasserstein loss. The real sequences $(\tau_1, \tau_2, \ldots, \tau_n, \ldots, \tau_{n+m})$ are randomly sampled from the real data pool. Our approach falls in the conditional generative model family because the generator's output is conditioned on the specific input: $(t_1, \ldots, t_n)$.

parameter w:

$$\min_\theta \max_w \underbrace{\sum_{l=1}^{L} f_w(\{\zeta_l, \rho_l\}) - \sum_{l=1}^{L} f_w(\{\zeta_l, g_\theta(\zeta_l)\})}_{\text{Wasserstein loss between two distribution}} - \underbrace{\lambda |f'_w(\hat{x}) - 1|}_{\text{1-Lipschitz regularizer}} \underbrace{- \sigma \log(P_\theta(\rho|\zeta))}_{\text{MLE loss}} \tag{8}$$

where $\hat{x}$ is the interpolation of $\{\zeta_l, \rho_l\}$ and $\{\zeta_l, g_\theta(\zeta_l)\}$ and the hyper-parameter λ is the regularization weight.

The working flow of our discriminator is illustrated in Figure 3. The overall learning procedure is given in Algorithm 1.

Experiments

Our experiment involves various public real-world datasets, and synthetic dataset simulated by popular point process models. We verify our approach on tasks for both prediction as well as the so-called cycle consistency check: train the generator using specified parametric models and its simulated data, and then learn the model parameters from the fake data generated by the trained generator.

Dataset and Implementation Details

Synthetic datasets. We simulate data from some classic and wildly-used point processes: inhomogeneous Poisson process (**IP**), self-exciting process (**SE**), i.e., Hawkes process in (Hawkes 1971), self-correcting process (**SC**) in (Isham and Westcott 1979). The detailed settings are specified as follows:

1. **Inhomogeneous Poisson process (IP).** The intensity function is independent from history and specified by a multi-modal function comprised of k Gaussian kernels:

$$\lambda(t) = \sum_{i=1}^{k} \alpha_i \left(2\pi\sigma_i^2\right)^{-1/2} \exp\left(-(t - c_i)^2/\sigma_i^2\right)$$

Algorithm 1 Conditional Wasserstein estimator (CWE) with MLE loss for event sequence prediction. The default values $\alpha = 1e{-}4, \beta_1 = 0.5, \beta_2 = 0.9, m = 256, n_{\text{critic}} = 5$.

Require: the trade-off coefficient σ, the regularization coefficient λ for direct Lipschitz constraint. the batch size, m. the number of iterations of the critic per generator iteration, n_{critic}. Adam hyper-parameters α, β_1, β_2.

Require: w_0, initial CNN discriminator f_w's parameters. θ_0, initial seq2seq LSTM generator g_θ's parameters.

while θ has not converged **do**
  for $t = 0, ..., n_{\text{critic}}$ **do**
    Sample $\{\zeta^{(i)}, \rho^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$ from real data.
    Generate $\{\zeta, g_\theta(\zeta)\}$.
    $\hat{x} = z\{\zeta, \rho\} + (1 - z)\{\zeta, g_\theta(\zeta)\}$ where $z \sim \text{Uniform}(0, 1)$.
    $L_w \leftarrow \sum_{l=1}^L (f_w(\{\zeta_l, \rho_l\}) - f_w(\{\zeta_l, g_\theta(\zeta_l)\})) - \lambda|f'(\hat{x}) - 1|$.
    $w \leftarrow \text{Adam}(\nabla_w L_w, w, \alpha, \beta_1, \beta_2)$
  end for
  Sample $\{\zeta^{(i)}, \rho^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$ from real data.
  $L_\theta = \frac{\sum_{i=1}^m f_w(\{\zeta_l, g_\theta(\zeta_l)\})}{m} - \sigma \log P_\theta(\rho^{(i)}|\zeta^{(i)})$
  $\theta \leftarrow \text{Adam}(-\nabla_\theta L_\theta, \theta, \alpha, \beta_1, \beta_2)$
end while

for $t \in [0, T)$ where c_i and σ_i are respectively fixed center and standard deviation, and α_i is importance weight of kernel i. Specifically, we set $k = 6, c = [3, 6, 9, 12, 15, 18], \sigma = [5, 5, 5, 5, 5, 5], \alpha = [14, 18, 13, 17, 10, 13]$.

2. **Self-exciting process (SE).** It is also called Hawkes process (Hawkes 1971) which assumes the past events increase the rate of future events. The effect decays over time captured by a decaying kernel g. Its conditional intensity function can be specified by

$$\lambda(t) = \mu + \beta \sum_{t_i < t} g(t - t_i)$$

where g is a nonnegative kernel function. Note μ is the exogenous rate of firing events and β is the coefficient for the endogenous rate. Here we specify $g(t) = \exp(-\omega t)$ for some $\omega > 0$, and set $\mu = 1.0, \beta = 0.8$ and the decaying kernel $g(t - t_i) = e^{-(t-t_i)}$.

3. **Self-correcting process (SC).** In this process, the intensity increases over time, while the past occurred event will decrease the occurrence probability in future. The conditional intensify function can be specified as:

$$\lambda(t) = \exp(\eta t - \sum_{t_i < t} \gamma).$$

The $\exp(\cdot)$ ensures that the intensity is positive, while η and γ are exogenous and endogenous rates. We set $\eta = 1.0, \gamma = 0.2$ in our experiment.

To further test the generalization ability of the proposed model in the case of predicting multi-modal data, we create 3 more datasets by a uniform mixture of the tuples above, namely **IP+SE**, **IP+SC**, **SE+SC**. They are used to testify the mode dropping problem of learning a generative model. For each synthetic dataset mentioned above, total $20,000$ sequences with each 60 events are simulated.

Real-world datasets. We evaluate our approach against peer methods on four real-world datasets from different domains: health-care records from MIMIC-III, job-hopping records from LinkedIn, IPTV users' watching records, and NYSE stock exchanges. Without loss of generality, the time scale for all real data are scaled to $[0, 15]$, and the details are as follows:

1. **MIMIC.** MIMIC-III (Medical Information Mart for Intensive Care III) is a large, publicly available dataset. It contains the health-related data of over $40,000$ anonymous patients from 2001 to 2012. We perform test on the patients who have at least three admission records, resulting in 2246 patients. Their admission timestamps are collected as event sequences. The dataset was downloaded from https://mimic.physionet.org.

2. **LinkedIn.** The LinkedIn data (Xu, Luo, and Zha 2017) is crawled online, which contains the job hopping records of over $3,000$ LinkedIn users in more than 80 Information Technology (IT) companies, research institutes and universities. Each user's onboard timestamps corresponding to different affiliations are recorded as an event sequence. The data can be found at https://github.com/HongtengXu/Hawkes-Process-Toolkit/blob/master/Data/LinkedinData.mat, and more details are available in (Xu and Zha 2017).

3. **IPTV.** The IPTV (Internet Protocol Television) log-data (Luo et al. 2014; 2015) is collected from a large-scale IPTV provider of Shanghai Telecomm Inc. The log-data records TV watching behaviors of $7,100$ users, which is composed of anonymous user logs, time stamps (at the precision of one second) of the beginnings and the endings of watching sessions (only those whose duration is more than 20 minutes are recorded), and TV programs which can be categorized into 16 classes. The data can be found at https://github.com/HongtengXu/Hawkes-Process-Toolkit/blob/master/Data/IPTVData.mat.

4. **NYSE.** The dataset contains 0.7 million high-frequency transaction records at a stock market in one day. The transactions are evenly divided into $3,200$ sequences with equal durations. The data can be found at https://github.com/dunan/NeuralPointProcess/tree/master/data/real/book_order.

The data for each type is divided into train, validation and test parts according to 0.7, 0.1, 0.2 ratio. The regularization term's weight λ for the 1-Lipschitz function f_w and the MLE loss's weight σ are determined by validation data. All temporal sequences are transformed into time duration between two consecutive events, i.e., transforming the sequence $\xi = \{t_1, \ldots, t_n\}$ into $\{\tau_1, \ldots, \tau_{n-1}\}$, where $\tau_i = t_{i+1} - t_i$. The transformed sequences are statistically identical to the original sequences, which can be used as the inputs of our neural network[1]. For synthetic data, a half of

[1] In fact, we can go further to embed the time duration into a low-dimensional vector feature, which is easy to handle for neural

events are taken as observations. For the three mixed synthetic data, the model's parameters is separately learned with their related generated sequences. For real data, sequences are padded with zeros to the same length and a half of each sequence are taken as observations. The implementation is based on TensorFlow and all experiments are executed on 12 Nvidia Tesla K80 GPUs.

Competitors and evaluation metrics

We compare our conditional Wasserstein estimator (**CWE**) with existing maximum likelihood-based methods on learning and predicting the parametric point processes mentioned in the previous subsection. Specifically, the competitors of our method include the MLE-based estimators of inhomogeneous Poisson process **(MLE-IP)**, self-exciting process **(MLE-SE)** and self-correcting process **(MLE-SC)**. Additionally, the recurrent network (Du et al. 2016) whose learning are all based on the traditional MLE loss (**MLE-NN**), the MLE based sequence-to-sequence model (**Seq2Seq**) (Sutskever, Vinyals, and Le 2014), and scheduled sampling model (**SS**) in (Bengio et al. 2015) are also considered.

We use all the above solvers to learn the model and predict forward sequences. For the synthetic data which we know the underlying generative models, we use the generated sequences to learn their parameters and compare our estimations with the ground truth. The deviation of learned parameters $\hat{\theta}$ and the ground truth $\theta^\star$ is defined as:

$$Pa.Dev. = \frac{||\hat{\theta} - \theta^\star||_2}{||\theta^\star||_2}. \tag{9}$$

Additionally, we can compare the error over the predicted sequence with the real sequence, which is defined as:

$$\begin{aligned} Pr.Dev. &= \|\xi - \eta\|_\star \\ &= \sum\nolimits_{i=1}^{n} |t_i - \tau_i| + \sum\nolimits_{i+1}^{m} T - \tau_{i+1}, \end{aligned} \tag{10}$$

where $\xi = \{t_1, \ldots, t_n\}$ is the predicted sequence and $\eta = \{\tau_1, \ldots, \tau_m\}$ is the real sequence. For the real-world data, we only use $Pr.Dev$ as the metric because the ground truth is unavailable.

Comparisons and analysis

Synthetic data. Table 1 shows the mean and standard deviation for 10 round runs of estimators and compares the learning ability of different estimator when the ground-truth data is generated via different types of point process. We first compare the deviation of parameters in the top row. The CWE generates higher quality sequences conditional on partial observation with no prior information about underlying process. Even compared with the estimators that do not suffer to model misspecification (i.e., the numbers with underlines in Table 1), CWE still has competitive performance. Whenever the model is misspecified (i.e., we don't know the parametric formulations or *right priori*) CWE outperforms other competitors, including those neural network-based ones (i.e., the bold numbers in Table 1). CWE performs better than Seq2Seq solved by MLE method justify the Wasserstein loss which consider the whole underlying structure between samples helps the generation process.

The second row of Table 1 compares the predicted sequences deviations. Similarly, the MLE-based parametric models can predict future events well in the case that we know the parametric formulations where the data comes from. Otherwise, the quality of generated sequences degrades considerably when the model is misspecified. We can observe our CWE produces better accuracy and performs robustly across different types of point process data.

To testify that CWE can deal with multi-modal data and cope with mode dropping, mixtures of data from three different point processes are designed. Models with specified form (e.g., IP, SE and SC) lack flexibility and fail to learn from diverse data sources. The third and forth row of Table 1 shows deviations of parameters and predicted sequences from the mixtures of data. CWE produces better sequences than its competitors, which fail to capture the heterogeneity in data.

In summary, our CWE method outperforms the other MLE-based methods except those having right prior knowledge of the parametric formulations. It should be noted that in most of practical situations, it is questionable to assume that the predefined models have no problem about model misspecification. From this viewpoint, CWE is superior in terms of performance and more feasible and robust to practical applications. The experiments on real-world datasets further demonstrate the superiority of our CWE method.

Real-world data. Table 2 shows the deviations between generated sequences and the ground truth. For real-world data that not necessarily obey the specific assumption by parametric point process models, CWE outperforms parametric point process models like MLE-IP, MLE-SE and MLE-SC by a notable margin because these competitors have high risks of model misspecification. Note the LinkedIn data has many doubly-censored short sequences, which makes the prediction harder as the training data is sparse and incomplete. Our CWE method still obtains encouraging prediction results in this case, which demonstrates its robustness to imperfect observations.

Moreover, CWE tends to outperform the Seq2Seq model which basically uses the same RNN architecture as ours but trained using MLE. This phenomenon is another evidence that considering Wasserstein distance-based loss is beneficial for us to learn a powerful generator. As aforementioned, MLE-based learning strategy may suffer from mode dropping or get stuck in a bad local optimum in the learning phase since maximizing likelihood is asymptotically equivalent to minimizing the Kullback-Leibler (KL) divergence between the data and model distribution. Adding the Wasserstein distance-based loss can mitigate this inherent pitfall.

Robustness to parameters

Finally, we use the data of self-exciting process to analyze the robustness of our CWE method to the 1-Lipschitz regularization term λ of function f_w and the trade-off term σ

networks. Now the time duration works well, we leave the time duration embedding for the future work.

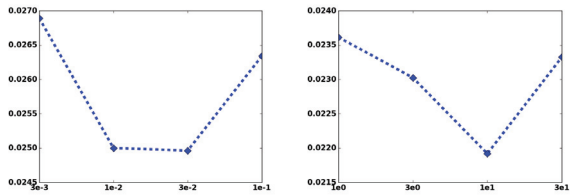
Table 1: Deviation of parameters and prediction for ground-truth and learned model by applying different methods on the synthetic data generated by different point processes.

	Model	Estimator						
		MLE-IP	MLE-SE	MLE-SC	MLE-NN	Seq2Seq	SS	CWE
Pa. Dev.	IP	**0.03 (3.0e-5)**	0.45 (5.0e-4)	0.67 (3.6e-4)	0.36 (3.5e-2)	0.31 (2.6e-3)	0.21 (5.6e-2)	0.09 (5.2e-3)
	SE	0.31 (4.6e-5)	0.02 (3.3e-4)	0.29 (1.5e-5)	0.24 (7.8e-3)	0.19 (2.3e-3)	0.15 (3.9e-2)	**0.02 (4.2e-3)**
	SC	0.94 (7.4e-4)	0.82 (7.4e-4)	**0.04 (8.8e-5)**	0.10 (2.6e-3)	0.12 (3.3e-3)	0.09 (3.5e-2)	0.07 (6.4e-3)
Pr. Dev.	IP	0.48 (1.3e-4)	0.79 (8.9e-5)	0.93 (3.4e-5)	0.72 (5.8e-2)	0.68 (6.6e-3)	0.64 (3.4e-2)	**0.45 (5.2e-3)**
	SE	1.55 (6.7e-5)	0.94 (1.9e-5)	1.52 (3.7e-4)	1.29 (4.5e-2)	1.27 (6.2e-3)	1.24 (8.2e-2)	**0.96 (9.1e-2)**
	SC	0.58 (7.3e-4)	0.76 (3.1e-5)	**0.33 (9.9e-4)**	0.44 (3.4e-3)	0.47 (5.2e-3)	0.40 (6.2e-3)	0.36 (6.3e-3)
Pa. Dev.	IP+SE	0.48 (6.2e-5)	0.36 (3.6e-5)	0.32 (6.7e-4)	0.23 (2.3e-2)	0.21 (3.4e-3)	0.18 (2.7e-2)	**0.08 (8.3e-3)**
	IP+SC	0.76 (5.3e-5)	0.88 (3.6e-4)	0.87 (6.2e-5)	0.28 (1.6e-2)	0.29 (7.5e-3)	0.23 (6.2e-3)	**0.11 (6.7e-3)**
	SC+SE	0.51 (7.2e-4)	0.69 (2.6e-4)	0.55 (6.3e-4)	0.32 (6.3e-2)	0.35 (7.5e-3)	0.29 (4.6e-2)	**0.15 (1.2e-3)**
Pr. Dev.	IP+SE	1.65 (5.4e-5)	1.41 (2.3e-5)	1.83 (5.3e-4)	1.03 (5.9e-2)	0.93 (3.1e-3)	0.89 (7.5e-2)	**0.76 (6.8e-3)**
	IP+SC	1.03 (3.0e-4)	0.98 (3.2e-4)	0.95 (0.9e-5)	0.43 (3.9e-3)	0.48 (6.2e-3)	0.40 (4.9e-3)	**0.31 (3.8e-3)**
	SC+SE	1.62 (4.5e-4)	1.43 (2.3e-5)	1.28 (6.7e-4)	0.89 (8.2e-2)	0.92 (4.6e-3)	0.85 (3.1e-2)	**0.63 (2.7e-3)**

Underlined numbers correspond to the results obtained when the real models are given.
Bold numbers correspond to the best results.

Table 2: Deviation of prediction for real-world data.

Data	Estimator						
	MLE-IP	MLE-SE	MLE-SC	MLE-NN	Seq2Seq	SS	CWE
MIMIC	0.25 (2.5e-5)	0.15 (5.3e-4)	0.26 (7.3e-5)	0.19 (2.3e-2)	0.17 (5.3e-3)	0.16 (4.1e-3)	**0.10 (2.5e-3)**
LinkedIn	0.24 (3.1e-4)	0.19 (4.8e-4)	0.17 (9.3e-4)	0.14 (9.1e-3)	0.14 (4.1e-3)	0.12 (8.9e-2)	**0.11 (9.4e-2)**
IPTV	1.46 (3.4e-5)	1.24 (2.8e-5)	1.52 (8.1e-5)	1.21 (2.8e-3)	1.19 (4.2e-2)	1.13 (8.4e-3)	**0.95 (4.9e-3)**
NYSE	2.25 (4.1e-5)	1.96 (6.5e-4)	2.34 (7.3e-5)	1.57 (4.8e-2)	1.55 (2.9e-3)	1.47 (7.3e-3)	**1.23 (2.8e-3)**



(a) sensitivity on λ (b) sensitivity on σ

Figure 4: The changes of $Pa.Dev$ with respect to (a) λ (fix $\sigma = 10$) and (b) σ (fix $\lambda = 0.01$) on the synthesized test data generated by self-exciting process.

between Wasserstein and MLE loss, respectively. In Figure 4(a), we can find that the performance of CWE to λ is less sensitive — only slight fluctuation of $Pa.Dev$ happen in a narrow range when changing λ in a wide range. This phenomenon happens across all datasets and coincides with the results in (Gulrajani et al. 2017), which verifies the robustness of our method to the selection of discriminator.

The appropriate σ keeps a balance between Wasserstein loss and MLE loss, which is more important for our method. In Figure 4(b), we can find that when σ is too small (i.e., < 10), the final results are degraded because the Wasserstein loss is dominant, which may lead to biased results. On the contrary, when σ is too high (i.e., > 10), the results become bad as well, which implies that the MLE loss is just a complementary component in complicated learning problems because merely considering the statistical similarity between high-dimensional distributions is not enough in these situations. Actually, Table 1 has shown that when just using MLE loss for Seq2Seq model which is equivalent to $\sigma = +\infty$, CWE performs better than MLE loss.

Conclusion and Future Work

We have developed an adversarial learning technique for (long-term) time dependent event sequence prediction. Our model can be seen as a conditional Wasseratein loss learning approach for event prediction. The proposed loss differs from the traditional MLE based one by adding a Wasseratein loss to measure the distance between two distribution realized by long-range event sequences. Extensive experiments on both synthetic and real-world data collaborate the efficacy of our method. In future work, online learning of the proposed model for being continuously updated for streaming data will be explored. Introducing the hidden Markov model (HMM) (Rabiner and Juang 1986) into our framework for high-dimensional event sequence learning is also an interesting venue for further study.

Acknowledgments

This project was supported in part by NKRDP 2016YFB1001003, NSFC 61602176, NSF (IIS-1639792, IIS-1218749, IIS-1717916, CMMI-1745382), NIH BIGDATA 1R01GM108341, NSF CAREER IIS-1350983, ONR

N00014-15-1-2340.

References

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

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875*.

Bacry, E.; Mastromatteo, I.; and Muzy, J.-F. 2015. Hawkes processes in finance. *Market Microstructure and Liquidity* 1(01):1550005.

Bellemare, M. G.; Danihelka, I.; Dabney, W.; Mohamed, S.; Lakshminarayanan, B.; Hoyer, S.; and Munos, R. 2017. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*.

Bengio, S.; Vinyals, O.; Jaitly, N.; and Shazeer, N. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *NIPS*, 1171–1179.

Choi, E.; Biswal, S.; Malin, B.; Duke, J.; Stewart, W. F.; and Sun, J. 2017. Generating multi-label discrete electronic health records using generative adversarial networks. In *arXiv preprint arXiv:1703.06490*.

Du, N.; Dai, H.; Trivedi, R.; Upadhyay, U.; Gomez-Rodriguez, M.; and Song, L. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *SIGKDD*, 1555–1564. ACM.

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

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. 2017. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*.

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

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

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

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2016. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*.

Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. *CVPR*.

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

Li, Y. 2017. Time-dependent representation for neural event sequence prediction. *arXiv preprint arXiv:1708.00065*.

Lian, W.; Henao, R.; Rao, V.; Lucas, J.; and Carin, L. 2015. A multitask point process predictive model. In *ICML*, 2030–2038.

Luo, D.; Xu, H.; Zha, H.; Du, J.; Xie, R.; Yang, X.; and Zhang, W. 2014. You are what you watch and when you watch: Inferring household structures from iptv viewing data. *IEEE Transactions on Broadcasting* 60(1):61–72.

Luo, D.; Xu, H.; Zhen, Y.; Ning, X.; Zha, H.; Yang, X.; and Zhang, W. 2015. Multi-task multi-dimensional hawkes processes for modeling event sequences. In *IJCAI*, 3685–3691.

Minor, B.; Doppa, J. R.; and Cook, D. J. 2015. Data-driven activity prediction: Algorithms, evaluation methodology, and applications. In *SIGKDD*, 805–814. ACM.

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

Press, O.; Bar, A.; Bogin, B.; Berant, J.; and Wolf, L. 2017. Language generation with recurrent generative adversarial networks without pre-training. *arXiv preprint arXiv:1706.01399*.

Rabiner, L., and Juang, B. 1986. An introduction to hidden markov models. *IEEE ASSP Magazine* 3(1):4–16.

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

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *NIPS*. 3104–3112.

Theis, L.; van den Oord, A.; and Bethge, M. 2015. A note on the evaluation of generative models. In *arXiv preprint arXiv:1511.01844*.

Xiao, S.; Farajtabar, M.; Ye, X.; Yan, J.; Song, L.; and Zha, H. 2017a. Wasserstein learning of deep generative point process models. In *arXiv preprint arXiv:1703.06490*.

Xiao, S.; Yan, J.; Farajtabar, M.; Song, L.; Yang, X.; and Zha, H. 2017b. Joint modeling of event sequence and time series with attentional twin recurrent neural networks. *arXiv preprint arXiv:1703.08524*.

Xiao, S.; Yan, J.; Yang, X.; Zha, H.; and Chu, S. M. 2017c. Modeling the intensity function of point process via recurrent neural networks. In *AAAI*, 1597–1603.

Xu, H., and Zha, H. 2017. Thap: A matlab toolkit for learning with hawkes processes. *arXiv preprint arXiv:1708.09252*.

Xu, H.; Wu, W.; Nemati, S.; and Zha, H. 2017. Icu patient flow prediction via discriminative learning of mutually-correcting processes. *TKDE*.

Xu, H.; Luo, D.; and Zha, H. 2017. Learning hawkes processes from short doubly-censored event sequences. In *ICML*.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, 2852–2858.

Zhang, Y.; Gan, Z.; Fan, K.; Chen, Z.; Henao, R.; Shen, D.; and Carin, L. 2017. Adversarial feature matching for text generation. In *ICML*.