Adversarial Training Model Unifying Feature Driven and Point Process Perspectives for Event Popularity Prediction

Conference Paper · October 2018

DOI: 10.1145/3269206.3271714

Chaoqi Yang
Tencent
4 PUBLICATIONS **6** CITATIONS
SEE PROFILE

Xiaofeng Gao
Shanghai Jiao Tong University
221 PUBLICATIONS **1,286** CITATIONS
SEE PROFILE

Paul Weng
Shanghai Jiao Tong University
67 PUBLICATIONS **358** CITATIONS
SEE PROFILE

Guihai Chen
Shanghai Jiao Tong University
398 PUBLICATIONS **5,981** CITATIONS
SEE PROFILE

Adversarial Training Model Unifying Feature Driven and Point Process Perspectives for Event Popularity Prediction

Qitian Wu
Shanghai Jiao Tong University
echo740@sjtu.edu.cn

Chaoqi Yang
Shanghai Jiao Tong University
ycqsjtu@gmail.com

Hengrui Zhang
Shanghai Jiao Tong University
sqstardust@sjtu.edu.cn

Xiaofeng Gao*
Shanghai Jiao Tong University
gao-xf@cs.sjtu.edu.cn

Paul Weng
Shanghai Jiao Tong University
paul.weng@sjtu.edu.cn

Guihai Chen
Shanghai Jiao Tong University
gchen@cs.sjtu.edu.cn

ABSTRACT

This paper targets a general popularity prediction problem for event sequence, which has recently gained great attention due to its extensive applications in various domains. Feature driven method and point process method are two basic thinking paradigms to tackle the prediction problem, but both of them suffer from limitations. In this paper, we propose PreNets unifying the two thinking paradigms in an adversarial manner. On one side, feature driven model acts like a 'critic' who aims to discriminate the predicted popularity from the real one based on a set of temporal features from the sequence. On the other side, point process model acts like an 'interpreter' who recognizes the dynamic patterns in sequence to generate a predicted popularity that can fool the 'critic'. Through a Wasserstein learning based two-player game, the training loss of the 'critic' guides the 'interpreter' to better exploit the sequence patterns and enhance prediction, while the 'interpreter' pushes the 'critic' to select effective early features that helps discrimination. This mechanism enables the framework to absorb the advantages of both feature driven and point process methods. Empirical results show that PreNets achieves significant MAPE improvement for both Twitter cascade and Amazon review prediction.

KEYWORDS

Event Sequence, Popularity Prediction, Adversarial Model, Feature Driven, Point Process

ACM Reference Format:

Qitian Wu, Chaoqi Yang, Hengrui Zhang, Xiaofeng Gao, Paul Weng, and Guihai Chen. 2018. Adversarial Training Model Unifying Feature Driven and Point Process Perspectives for Event Popularity Prediction. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3269206.3271714

1 INTRODUCTION

Event sequences are ubiquitous in various domains and can model many real-life phenomenons, such as retweet cascade of one Twitter post, purchase behaviors of one Amazon product, and infection of one epidemic disease. For Twitter post, event sequences record the occurring time of each retweet. For Amazon product, event sequences record the time each purchase or click behavior happens. For epidemic disease, event sequences consist of the time each infection takes place. Given an observed event sequence with limited length, one would usually like to know how many events would occur in the end, i.e., the *event popularity*. In social networks, predicting the popularity of an original tweet can help conduct rumor monitoring [38] and anomaly detection [6]. In e-commerce, predicting the popularity of a new product can be used for personalized recommendation [19] and targeted advertisement [15]. In the health-care domain, if one can predict the influence of an epidemic disease, then preventive measures could be taken to control the infection evolution. Building an effective model to predict event popularity can bring many economic and social benefits. Fig. 1 shows a flowchart which compares the two mainstream thinking paradigms that are used to tackle the prediction problem.

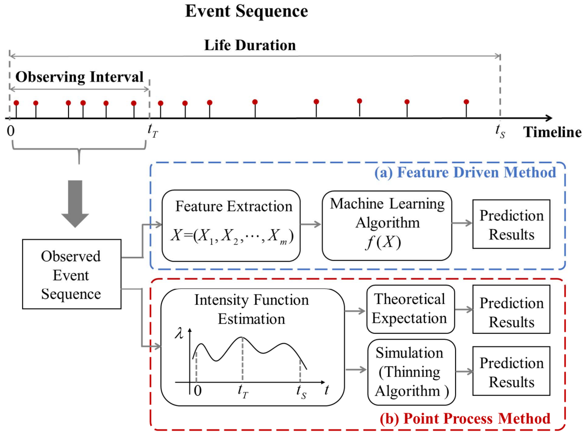

Figure 1: Two methods for event popularity prediction.

One thinking paradigm from a macroscopic view is feature driven method (Fig. 1(a)). As the name implies, this method first extracts a set of early features from observed event sequence and then leverages some machine learning algorithms to determine a mapping function from early features to popularity. Various features from different perspectives have been proved to be good early

*Xiaofeng Gao is the corresponding author.

CIKM '18, October 22–26, 2018, Torino, Italy
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3271714

popularity indicators, such as user profiles [24], user history activities [41], social network properties [10], and temporal characteristics of observed sequences [10, 16, 31]. While feature driven method can achieve decent accuracy and possess good robustness, it suffers from two limitations: i) *high human expertise*: feature selection requires a deep knowledge of a domain and lots of human intervention; ii) *high feature cost*: some models leverage user profiles or social network information, which could be inaccessible in practical scenarios due to some privacy issues.

Another thinking paradigm, point process method, uses a more principled approach (Fig. 1(b)). This method views an event sequence as a counting process and models the time intervals between events as random variables. One way to characterize the point process is by the conditional intensity function designed to model the time of next event occurrence given the history event sequence. Different forms of intensity functions can capture different dynamic patterns in the sequences, like inhomogeneous Poisson Process [30], Weibull model [42], and Hawkes Process [4, 18, 27, 43]. Based on the observed sequence, the intensity function can be estimated via Maximum Likelihood Estimation (MLE). After that, there are two ways to predict the popularity (in Fig. 1(b)). One is to derive the theoretical expectation of the event number when time goes to infinity [22, 43]. The other is to simulate the occurrence of future events by the thinning algorithm [9, 23]. While the point process method is equipped with solid theoretical foundations [7], it also has some limitations: i) *long observation dependency*: the precision highly relies on sufficient observation, which disqualifies it to accommodate early-stage prediction with short observed sequences; ii) *model failure*: the expectation of event number may be infinite in a supercritical region [43], which makes the model output an invalid predicted value.

Due to the limitations of feature driven and point process methods, solely adopting one method is not a satisfying solution, especially when the following two situations are encountered: i) early-stage prediction is needed, and ii) other features from social networks or user profiles are inaccessible. In this paper, inspired by Generative Adversarial Nets (GAN) [12] and its Wasserstein variant called WGAN [3], we propose PreNets (Adversarial Nets for Event Popularity Prediction) to combine the feature driven and point process methods through an adversarial training rule. On one side, point process model plays the role of what we call an 'interpreter', recognizing the patterns hidden in observed event sequence to generate the predicted popularity. On the other side, feature driven model plays as a 'critic', firstly extracting a set of temporal features from observed sequences and then using them to discriminate the predicted popularity given by the 'interpreter' from the real one. The Earth Moving (EM) distance, also known as Wasserstein distance [3], is adopted to measure the discrepancy between the predicted value and the ground-truth label.

The training stage is a two-player minimax game. The 'critic' aims at maximizing the EM distance based on the features, and then guides the training of the 'interpreter'. The 'interpreter' struggles to minimize the EM distance by giving a convincing predicted value, and then pushes the 'critic' to the limits. As the 'critic' becomes stronger, the 'interpreter' can obtain higher-quality guidance. This mechanism enables PreNets to take advantage of both thinking paradigms: i) point process model exploits the patterns hidden in sequence; ii) feature driven model leverages good early features to suit the model to early-stage prediction.

We apply PreNets to two real-world prediction tasks: retweet cascade on Twitter and movie review on Amazon. To validate the advantages of adversarial training, we show that PreNets could achieve better precision than independently trained point process and feature driven models. For comparison, we let PreNets compete with state-of-the-art models, and the results demonstrate the superiority of PreNets. Concretely, for Twitter cascade/Amazon review with short and long observation, PreNets respectively achieves 5.6%/7.2% and 21.1%/6.3% improvement on MAPE.

2 PRELIMINARY

Event sequence triggered by one source consists of a series of timestamps that record each event occurrence. For example, an attractive tweet on Twitter may cause a long retweet cascade, a popular product on Amazon could trigger streaming purchase or click behaviors, one epidemic disease would bring about an infection outbreak. Here, the tweet, the product, and the disease are called sources, while the retweet cascade, the user behaviors, and the infection process would form event sequences. For each case, one event happens when: i) a user forwards the tweet, ii) a user purchases the product or clicks on the webpage, iii) a new patient is infected by the disease. Here is a formal definition.

Definition 1. *(Event Sequence): For source i, we define the time sequence $\mathcal{S}^i(t) = \{t_j^i\}$ as event sequence up to time t, $1 \leq j \leq N^i(t)$. Here $\{t_j^i\}$ is the timestamp when event j happens and $N^i(t) = |\mathcal{S}^i(t)|$ denotes the event number for source i up to time t.*

Definition 2. *(Popularity): For source i, we define its popularity as $N^i(\infty) = \lim_{t\to\infty} |\mathcal{S}^i(t)|$. There exists t_s^i for source i such that $N^i(t_s^i) \approx N^i(\infty)$, and we call $[0, t_s^i]$ the life duration of source i.*

Point Process Theory. In point process theory, the time intervals between events are modeled as random variables. One point process can be characterized via conditional intensity function:

Definition 3. *(Conditional intensity function): For source i, $\lambda^*(t)$ is the expected instantaneous rate of new event occurrence given the history sequence:*

$$\lambda^*(t) = \lambda(t|\mathcal{S}^i(t)) = \frac{\mathbb{P}\{N^i(t+dt) - N^i(t) = 1|\mathcal{S}^i(t)\}}{dt}. \quad (1)$$

Here the notation $*$ serves as a reminder that the intensity is conditional on the history sequence.

In a small time interval $[t, t+dt)$, $\lambda^*(t)dt$ is the probability for a new event occurrence given the history sequence $\mathcal{S}^i(t)$. Then the conditional density function $f^*(t)$ for the time of next event since time t_n can be specified by[1]

$$f^*(t) = \lambda^*(t)\exp(-\int_{t_n}^{t} \lambda^*(\tau)d\tau). \quad (2)$$

Some particular forms of intensity functions are designed to capture different dynamic patterns in the sequence, such as inhomogenous Poisson Process [30], Weibull model [42], Hawkes Process with exponential kernel [4], power-low kernel [43], or Fourier based kernel [18].

[1] Please refer to [7] or a recent tutorial [26] for derivation.

Problem Formulation. The problem in this paper can be formulated as: for a new source i, given the observed event sequence $\mathcal{S}^i(t_T)$ (we call $[0, t_T]$ as the observing interval), we aim to predict its popularity $N^i(\infty)$. This problem is non-trivial due to the following two key points. 1) *Early-Stage Prediction*: The observing interval can be significantly short compared with the life duration of one event sequence. 2) *Poor Information Driven*: The available information is only limited to the source description and the observed sequence. Other features from user profiles, user history activities, social networks and some cross-platform information are unknown.

In the following parts, for compactness, we use $\mathcal{S}^i$, N^i, and n^i to simplify $\mathcal{S}^i(t_T)$, $N^i(\infty)$, and $N^i(t_T)$ respectively.

3 METHODOLOGY

In this section, we present our model PreNets that elegantly tackles the non-trivial prediction problem. Firstly, we cast a glance at an overview of the framework. Then we put forward the point process model and the feature driven model, which respectively plays as an 'interpreter' and a 'critic' in PreNets. Afterwards, we zoom in on the adversarial training for unifying two basic models by a minimax game.

3.1 Model Overview

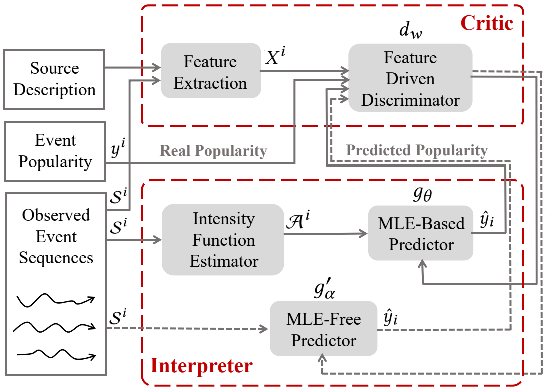

Figure 2: PreNets framework.

Fig. 2 is the framework of PreNets consisting of two parts: point process model ('interpreter') and feature driven model ('critic'). The inputs of PreNets are event sequence and its source description. Then the trained PreNets will output the predicted popularity for each event sequence.

The training procedure of PreNets is based on a minimax game between the 'interpreter' and the 'critic'. The 'interpreter' aims at using the observed sequence to predict its popularity. We propose two self-contained models to accomplish this mission from different perspectives. One is MLE-based point process model and the other is MLE-free point process model. For MLE-based model, we first use each event sequence to estimate an intensity functions (implemented by RNN) via maximum likelihood estimation (MLE). The estimated intensity functions can capture the dynamic patterns in the sequential data, and will be further input into a MLE-based predictor (implemented by another RNN). The predictor will output the predicted popularity. Differently, the MLE-free model skips the intensity function estimator, directly inputs the event sequences into the MLE-free predictor (implemented by RNN) and conducts prediction.

On the other side, feature driven model first extracts a set of features from each event sequence and basic information about the source (e.g., the posting time of one original tweet, the category of one product or the description of one epidemic disease). Then these features, the real popularity of each sequence as well as the predicted popularity given by the predictor will be input into a feature driven discriminator (implemented by a 3-layer ELU neural network), which is designed to distinguish the predicted value from the real one. Once we update the feature driven discriminator, the loss generated from the predicted popularity will be transferred back to the 'interpreter' as its feedback training signal. Afterwards, the predictor will be updated based on this feedback signal.

During the training stage, the 'interpreter' struggles to generate the popularity which can fool the 'critic', while the 'critic' tries its best to conduct the discrimination and then instructs the 'interpreter'. The game will be over once an equilibrium is reached. Then, the 'critic' can no longer distinguish the predicted popularity, and the 'interpreter' becomes a strong predictor. The framework absorbs the strengths of both feature driven and point process methods.

3.2 Interpreter: Point Process Model

The point process model aims at leveraging the observed sequence to predict its popularity. We propose two self-contained models to accomplish the prediction from different perspectives. In the following subsections, we will present our model design, its rationale, and the corresponding motivations.

3.2.1 *MLE-based Point Process Model.*

As introduced in Section 2, one way to characterize the point process is via the conditional intensity function. A slew of prior works assume specific parametric forms for the intensity function, such as inhomogeneous Poisson Process or Hawkes Process. Some of them are claimed to be effective in some specific tasks. For instance, Hawkes Process with power-low kernel [22, 43] works pretty well when predicting the information cascade in social networks. While the parametric assumption guarantees the accuracy for one kind of event sequence, it limits the model generality to other kinds of event sequence.

Recurrent Neural Network (RNN) is a feedforward neural network architecture where the current state depends on both the current input signal and the previous state. Such mechanism enables the network to memorize the influence of history signals. RNN has been shown to be powerful in modeling sequential data, such as handwriting recognition [13], and machine translation [33]. [8, 39, 40] leverage RNN to model the intensity functions of point process and achieve state-of-the-art performance.

Following the frontier researches [8, 40], we relax the parametric assumption for the intensity functions and leverage RNN to approximate the conditional intensity for event sequence.

Intensity Function Estimation. Considering one point process for source i, given the observed sequence $\mathcal{S}^i = \{t^i_j\}$. Use $\boldsymbol{h}_j$ to denote the hidden state of RNN, and we update the hidden layer once receiving a new event,

$$\boldsymbol{h}_{j+1} = \max\left\{\boldsymbol{w}_h \cdot t^i_j + \boldsymbol{v}_h \cdot \boldsymbol{h}_j + \boldsymbol{b}_h, 0\right\}. \tag{3}$$

The conditional intensity function based on $\mathcal{S}^i(t^i_j)$ can be specified as

$$\lambda^*(t) = \exp(\boldsymbol{v}_t \cdot \boldsymbol{h}_j + w_t(t - t^i_j) + b_t), \tag{4}$$

where in the exponential function, the first term aggregates the influence of previous events, the second term captures the current influence by event j, and the last term stands for the base intensity. The outside exponential function acts as a non-linear transformation [8].

Then according to (2), the log-likelihood of a collection of sequences $C = \{\mathcal{S}^i\}$, where $\mathcal{S}^i = \{t^i_j\}^{n^i}_{j=1}$, can be derived as

$$\begin{aligned} l(C) &= \sum_i \sum_j \log f^*(t_{j+1}) \\ &= \sum_i \sum_j \mathbf{v}_t \cdot \mathbf{h}_m + w_t(t^i_{j+1} - t^i_j) + b_t + \frac{1}{w_t}\exp(\mathbf{v}_t \cdot \mathbf{h}_m + b_t) \\ &\quad - \frac{1}{w_t}\exp(\mathbf{v}_t \cdot \mathbf{h}_m + w_t(t^i_{j+1} - t^i_j) + b_t). \end{aligned} \tag{5}$$

By maximizing the log-likelihood, one can obtain the intensity functions $\lambda^*(t)$ for each source i. We use stochastic gradient descent as our optimization algorithm.

Popularity Prediction. We further probe into the prediction. Based on the intensity functions, we concatenate the intensity values with the original event timestamps and define an intensity sequence:

$$\mathcal{A}^i = \left\{A^i_j\right\}^{n^i}_{j=1}, \quad A^i_j = (t^i_j, \lambda^*(t^i_j)). \tag{6}$$

The intensity sequence records the timestamp of each event and the corresponding intensity value, which can reflect the dynamic patterns in one sequence. So one can consider $\mathcal{A}^i$ as a kind of representation for the observed event sequence.

Then we again adopt RNN to exploit the relationship between intensity sequence $\mathcal{A}^i$ and the popularity N^i, i.e., $g_\theta : \mathcal{A}^i \to N^i$. In our experiment, we also leverage more complex architecture, Long Short-Term Memory (LSTM) network, to implement g_θ, but here we use RNN to give a more concise illustration. Use $\boldsymbol{h}_j$ to denote the hidden state and $\hat{y}$ to denote the output result, we have

$$\boldsymbol{h}_{j+1} = \tanh(\boldsymbol{v}'_h \cdot A^i_j + \boldsymbol{w}'_h \cdot \boldsymbol{h}_j + \boldsymbol{b}'_h), \tag{7}$$

$$\hat{y} = \boldsymbol{w}_o \cdot \boldsymbol{h}_M + b_o, \tag{8}$$

where $M = \max n^i$ and the model parameter set $\theta = \{\boldsymbol{v}'_h, \boldsymbol{w}'_h, \boldsymbol{b}'_h, \boldsymbol{w}_o, b_o\}$. Then the problem remains as how to train g_θ. We will focus on it in Section 3.4.

The MLE-based point process model first converts the event sequence to intensity sequence as a representation and then uses the intensity sequence to predict popularity. To obtain the intensity functions, one must optimize the log-likelihood function. Solving the MLE requires high computational complexity in practice, so we propose another point process model.

3.2.2 MLE-Free Point Process Model. Since our goal is to predict the popularity rather than to understand the patterns in the event sequence, we can skip the estimation of intensity functions and the MLE, directly targeting the relationship between observed sequence $\mathcal{S}^i$ and the popularity, i.e., $g'_\alpha : \mathcal{S}^i \to N^i$ by RNN. Concretely,

$$\boldsymbol{h}_{j+1} = \tanh(\boldsymbol{v}''_h \cdot t^i_j + \boldsymbol{w}''_h \cdot \boldsymbol{h}_j + \boldsymbol{b}''_h), \tag{9}$$

$$\hat{y} = \boldsymbol{w}'_o \cdot \boldsymbol{h}_M + b'_o. \tag{10}$$

The model parameter set $\alpha = \{\boldsymbol{v}''_h, \boldsymbol{w}''_h, \boldsymbol{b}''_h, \boldsymbol{w}'_o, b'_o\}$.

To put it more symbolically, each point process model is like an 'interpreter', who works on interpreting the event sequence into a predicted popularity value. The MLE-based point process model first converts the sequence into an intensity sequence, and then conducts prediction, while the MLE-free model tackle the prediction directly. Experiments in Section 4 sheds more insights by comparison between these two models.

3.3 Critic: Feature Driven Model

Unlike point process model, feature driven method studies the prediction problem from the perspective of feature extraction. In our PreNets, different from previous works, we leverage feature driven model to discriminate the predicted popularity given by the 'interpreter' from the real one.

Feature Extraction. Extracted features should reflect the characteristics of observed event sequence from various aspects. The features used in this paper includes: number of events in the first/ second half of observing interval, increment of events in the first/ second half of observation interval, mean time interval between events in the first/second half of observing interval, maximum time interval between events, mean and standard deviation of event number per time unit, mean and standard deviation of time interval between events per time unit, five point summary of waiting time between events, five point summary of event number per time unit, coefficients of polynomial curve fitting for the cumulative event number. Note that all the features mentioned above can be easily obtained given the observed event sequence $\mathcal{S}^i$.

These features are proved to be effective early indicators of popularity in previous papers, and here we provide a brief justification for some of them. The volume of events in observing interval reflects the early influence of the source [16]. The time interval between events captures the early attractiveness of the source [31]. The increment of events characterizes the popular potentiality [16]. The maximum time interval is related to the possible dormant cycle [10]. The five point distribution statistics approximately describe the temporal similarity among different event sequences, while the fitting curve shows the basic evolving trending of event number.

Moreover, some descriptive information about the source can also be leveraged as features, such as the generation time of the source (e.g., the posting time of the original tweet) and the category of the source (e.g., the product category or the disease category). These source features may vary for different tasks.

Some related studies rely on other information, like user profiles, history records of user activities, and social network properties. While such features are claimed to be important early indicators of popularity, they tend to be inaccessible in practice due to privacy concerns. Hence, to make our model equipped with good generality, we solely rely on source features and temporal features that can be easily extracted from observed event sequences. For a more thorough study, we will extend our model to such a case with more features in experiment section and study the performance improvement brought by more early features.

Loss Function. The Earth Moving (EM) distance [3], also called Wasserstein distance, measures the distance between two probability distributions. Our model adopts EM distance to quantify the discrepancy between the predicted popularity and the real one to enable the feature driven model to conduct discrimination.

For one source i, we define two vectors $\eta^j = (X^i, y^i)$ and $\zeta^i = (X^i, \hat{y}^i)$, where X^i contains the features for i, y^i is the real popularity value, and $\hat{y}^i$ is the predicted popularity given by point process model. Assume η obeys a probability distribution $\mathbb{P}_r$ and ζ obeys a probability distribution $\mathbb{P}_g$. The Earth Moving (EM) distance between two probability distributions can be defined as follows:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\phi \sim \Phi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(\eta, \zeta) \sim \phi}[||\eta - \zeta||], \tag{11}$$

where $\Phi(\mathbb{P}_r, \mathbb{P}_g)$ denotes the set of all joint distributions $\phi(\mathbb{P}_r, \mathbb{P}_g)$ whose marginal are $\mathbb{P}_r$ and $\mathbb{P}_g$. (11) can be equivalently written as:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \sup_{||d||_L \leq 1} \mathbb{E}_{\eta \sim \mathbb{P}_r}[d(\eta)] - \mathbb{E}_{\zeta \sim \mathbb{P}_g}[d(\zeta)], \tag{12}$$

where the supremum is taken over all Lipschitz functions $d : \Omega \rightarrow \mathbb{R}$ and Ω denotes the value space for η and ζ.

(12) is hard to solve since we can not enumerate all Lipschitz functions. Instead, we assume d is parametrized with w and use a specific function form to approximate the search space on all Lipschitz functions. Then the problem boils down to solving

$$\max_{w, ||d_w||_L \leq 1} \mathbb{E}_{\eta \sim \mathbb{P}_r}[d_w(\eta)] - \mathbb{E}_{\zeta \sim \mathbb{P}_g}[d_w(\zeta)]. \tag{13}$$

The more flexible form of d_w will give more accurate approximation, and we will leave its discussion in our experiments.

In fact, the output result of $d_w(\cdot)$ measures the confidence level of popularity value given the features (the higher $d_w(\cdot)$ means the more confident), and (13) measures the discrepancy between the confidence levels of predicted popularity and the real one. The maximum in (13) is to enlarge the distance between probability distributions of predicted popularity and the ground-truth label conditioned on extracted features, which contributes to a fine discrimination. So the feature driven model is like a 'critic', using features to judge if the predicted value is convincing. (Indeed, the 'critic' notion is used to describe discriminator in WGAN [3].)

3.4 Adversarial Training

In PreNets, we let the point process model and the feature driven model be trained in an adversarial way, based on Wasserstein learning proposed by [3]. The feature driven model aims at discriminating the predicted popularity given by the point process model from the real popularity, i.e., maximizing the EM distance (see (13)). The point process model tries to fool the feature driven model by giving a convincing predicted result, i.e., minimizing the EM distance. So the global training objective can be written as:

$$\begin{aligned}\min_{\theta} \max_{w, ||d_w||_L \leq 1} & \mathbb{E}_{(X,y) \sim \mathbb{P}_r}[d_w(X, y)] \\ & - \mathbb{E}_{(X, g_\theta(\mathcal{A})) \sim \mathbb{P}_g}[d_w(X, g_\theta(\mathcal{A})], \end{aligned} \tag{14}$$

where $g_\theta(\mathcal{A}) = \hat{y}$ denotes the predicted result given by (7) and (8) of RNN[2]. (14) can be approximated by solving

$$\min_{\theta} \max_{w, ||d_w||_L \leq 1} \frac{1}{B} \sum_{i=1}^{B} [d_w(X^i, y^i)] - \frac{1}{B} \sum_{i=1}^{B} [d_w(X^i, g_\theta(\mathcal{S}^i)], \tag{15}$$

where B is the batch size. Following [39], we further convert the Lipchitz constraints into the gradient regularization in the objective:

$$\begin{aligned}\min_{\theta} \max_{w} & \frac{1}{B} \sum_{i=1}^{B} [d_w(X^i, y^i)] - \frac{1}{B} \sum_{i=1}^{B} [d_w(X^i, g_\theta(\mathcal{S}^i)] \\ & - \gamma \sum_{i,j=1}^{B} \Big| \frac{|d_w(X^i, y^i) - d_w(X^j, g_\theta(\mathcal{S}^j)|}{|y^i - g_\theta(\mathcal{S}^j)|} - 1 \Big|, \end{aligned} \tag{16}$$

where γ is the weight for regularization.

Then we provide the training algorithm for PreNets in Alg. 1. Before conducting adversarial training, we pretrain d_w and g_θ. When pretraining g_θ, we minimize the mean square loss

$$\min \frac{1}{B} \sum_{i=1}^{B} (g_\theta(\mathcal{S}^i) - y^i)^2. \tag{17}$$

Then during the adversarial training stage, the feature driven model and the point process model are updated alternatively until convergence. Note that Alg. 1 is designed for MLE-based PreNets (MLE-based point process model v.s. feature driven model), and line 7, line 13 will be removed for MLE-free PreNets (MLE-free point process model v.s. feature driven model).

Algorithm 1: Minimax Game for PreNets

```
1  REQUIRE: $\gamma$, the weight for regularization. $n_{critic} = 6$, the number of
   iterations of the 'critic' per 'interpreter' iteration. $B = 64$, the batch size.
   $\alpha = 5e-4$, $\beta_1 = 0.9$, $\beta_2 = 0.9$, Adam hyper-parameters.
2  REQUIRE: $w_0$, initial 'critic' parameter. $\theta_0$, initial 'interpreter' parameter.
3  Pretrain $d_w$ and $g_\theta$ using data $C = \{\mathcal{S}^j\}$;
4  while not converged do
5      for $k = 0, \cdots, n_{critic}$ do
6          Sample $\{(X^i, y^i)\}_{i=1}^B \sim \mathbb{P}_r$ from real data;
7          Compute $\lambda^*(t)$ by (4) and get $\mathcal{A}^i$ by (6) for each $i$;
8          $L_C \leftarrow \frac{1}{B} \sum_{i=1}^{B} [d_w(X^i, g_\theta(\mathcal{S}^i)] - \frac{1}{B} \sum_{i=1}^{B} [d_w(X^i, y^i)]$
9          $+\gamma \sum_{i,j=1}^{B} \Big| \frac{|d_w(X^i, y^i) - d_w(X^j, g_\theta(\mathcal{S}^j)|}{|y^i - g_\theta(\mathcal{S}^j)|} - 1 \Big|$;
10         $w \leftarrow$Adam$(\nabla L_C, \alpha, \beta_1, \beta_2)$;
11     end
12     Sample $\{(X^i, y^i)\}_{i=1}^B \sim \mathbb{P}_r$ from real data;
13     Compute $\lambda^*(t)$ by (4) and get $\mathcal{A}^i$ by (6) for each $i$;
14     $L_I \leftarrow \frac{1}{B} \sum_{i=1}^{B} [f_w(X^j, g_\theta(\mathcal{S}^i)]$;
15     $\theta \leftarrow$Adam$(-\nabla L_I, \alpha, \beta_1, \beta_2)$;
16 end
```

The adversarial training is a two-player game between the 'interpreter' and the 'critic'. They are struggling to 'beat' each other, through which the potential energy of them can be exploited: i) the 'interpreter' could better recognize the patterns hidden in sequence; ii) the 'critic' could select effective early features that help discrimination. The game will terminate when an equilibrium is

[2] The derivation is designed for MLE-based point process model. For MLE-free model, one can replace the $g_\theta(\mathcal{A})$ by $g'_\alpha(\mathcal{S})$.

reached [12], where the 'critic' can no longer discriminate the predicted popularity and the 'interpreter' can provide prediction which synthesizes the information contained in sequence patterns and the temporal features. Hence, the framework absorbs the strengths of both point process and feature driven methods. Our experiments will validate the advantages of adversarial training over independently training g_θ and d_w.

3.5 Links to Related Work

In order to shed more lights on our methodology, we compare PreNets with some existing works from three-fold perspectives.

3.5.1 Popularity Prediction. Most early studies on event popularity prediction are based on feature driven method. [32], as one of the pioneer setting foot in predicting single tweet popularity, uses generalized linear model to find the relationship between some early features and the retweet count. In addition, a series of works have tried various features from user profiles [24], history activity records [41], social networks [10], and some cross-platform information [34]. Afterwards, point process models [4, 30, 43] are proposed to tackle the prediction problem from a more principled way. With good interpretability and solid theoretical foundations, point process models have been widely adopted by many researchers [9, 27, 29, 37]. However, these two methods suffer from respective limitations as discussed in Section 1.

Moreover, the early studies on popularity prediction rely heavily on long-term observation. Motivated by this, some recent researches target the early-stage prediction by probabilistic model [21] and regression tree [22]. However, they leverage some other information from social networks [21] or user profiles [22] to make up for the information poverty led by insufficient observation.

Compared with previous works, our PreNets owns the following advantages: i) *Model Unification*: PreNets unifies point process and feature driven models through a minimax game, absorbing the strengths of both two thinking paradigms. ii) *Generality*: Our point process model makes no parametric assumption for the intensity functions and our feature driven model only relies on features that can be directly extracted from the event sequences. These two facts enable our model to be generalized to any prediction task for sequential data, while most existing works target only one or some kinds of event sequence and make several task-specific assumptions or designs. iii) *Low Feature Cost*: Our model does not leverage other information such as user profiles or social network information, which are claimed to be indispensable in some other works. Our experiments show that PreNets is competent in early-stage prediction even with only temporal information.

3.5.2 Hybrid Prediction Model. Several recent studies make an attempt to incorporate feature driven and point process perspectives in one prediction model. [35] and [11] leverage survival theory to combine features with the point process model. The features are used to describe some characteristics of the events, and are input into the survival model together with the event timestamps. [22] proposes a hybrid model, in which the MLE-based Hawkes Process model is used to conduct a first-layer estimation of the popularity and then the features are applied to add an adjusting weight on the first-layer estimation to refine the precision. The results show that the hybrid model outperforms some state-of-the-art models for information cascade prediction. [20] proposes a lifetime aware regression method to predict the popularity of YouTube video. Some early features are used to determine the life spans of videos, based on which the videos would be divided into different groups. Videos in one group share the same parametric form of the regression function.

In contrast with them, our PreNets is quite different in the following four aspects. First, in PreNets, the feature driven model and the point process model are two self-contained models, instead of combined into a new model like [11, 35] or concatenated as two layers of one model [20, 22]. Second, we leverage adversarial training to unify two models and the objective is to search an equilibrium of the minimax game, instead of solving a MLE function or minimizing the mean square error. Third, in the adversarial training stage, the goal of the feature driven model is to discriminate the predicted value from the real one, so the features are used to help the 'critic' conduct discrimination rather than prediction. Also, whether [22] or [20] uses a lot of features from user profiles and social networks, which are not mandatory in PreNets.

3.5.3 Adversarial Training. Generative Adversarial Nets (GAN) has shown its great power in image and text generation as well as semi-supervised learning. The original GAN in [12] aims at minimizing the Jensen-Shannon divergence but is shown to suffer from highly instability and collapse mode [2, 28]. To refine the learning of GAN, [3] proposes Wasserstein GAN (WGAN) which adopts the Earth Moving (EM) distance as the objective for training. It is proved that using EM distance as the metric between probability distributions possesses many advantages, including reducing the mode dropping [14] and refining the generated samples. Our work extends Wasserstein learning to event popularity prediction problem but there are some differences. Firstly, the point process model aims at predicting the popularity from the observed sequence, instead of generating a complete event sequence based on the input of random data (like [39]) , which can be viewed as a counterpart for image generator using GAN. Hence, we call our point process model as an 'interpreter' throughout the paper, instead of a generator. The other difference lies in the feature driven model. In original GAN and WGAN, the generated pictures by the generative model would be the input of the discriminative model as the negative samples, while in PreNets, the features of one sequence are from the real sequence data, rather than the point process model.

There have been lots of successful applications of GAN in many domains, such as information retrieval [36], text to image synthesis [25], dialogue generation [17] and cryptography [1]. The proposal of GAN paves a way to unify two different models in an adversarial manner. Compared with other GANs unifying the generative and discriminative models, more precisely, PreNets unifies feature driven and point process models. In other words, PreNets is custom-made for event popularity prediction problem.

4 EXPERIMENTS

In this section, we apply our PreNets to two real-world prediction tasks : *retweet cascade* on Twitter and *movie review* on Amazon. The difference between two tasks lies in the different dynamic patterns of the event sequences. For retweet cascade, it is proved that the previous retweet behaviors may have a strong correlation with the future retweet [5], since the users in social networks are prone to be influenced by the posts of whom they follow. For movie review, this correlation is not that significant. Another difference is that the life durations for most retweet cascades are one or two days, while the review sequence for one movie may span several months.

In the following parts, we will first introduce the experiment setup including the data set information, the evaluation protocol, and the comparative methods. Then we focus on the experiment results and do some discussions.

4.1 Experiment Setup

4.1.1 Date Sets. Our experiments are conducted on two large data sets.

Twitter. We use the public data provided by [43] to construct our Twitter data set. This data set contains roughly 160 thousand retweet cascades in total. For each cascade, it records the timestamps of each tweet post. We first filter the cascades with length between 50 and 5000. Then since our shortest observation time is five minutes, we drop the cascades whose number of tweets posted in the first five minutes are less than 10. Finally, we construct totally 21,890 cascades for our experiments.

Amazon. This data set is provided by Stanford Large Network Dataset Collection. The data ranges from Aug. 1997 to Oct. 2012, including approximately 8 million reviews, 250 thousand movies and 890 thousand users. Also, we first filter the movies with review number between 50 and 8000. Then we only consider the movies whose number of reviews in the first twenty days are more than 10. After that, we obtain 32,134 movies. This data set also provides user information and the plaintext of reviews. We basically ignore this extra information and will further discuss the improvement brought by incorporation of these features.

4.1.2 Evaluation Protocol. For each data set, we randomly choose 80% event sequences as training set and the remaining sequences as test set. We basically set two different observation times for comparison between prediction with short observation (early-stage prediction) and long observation. Concretely, for Twitter cascade, we consider 10 minutes and 1 hour observation. For Amazon review, we consider 20 days and 100 days observation. To study the impact of different observation time, we will probe into the performance variation as the observation time increases gradually.

We adopt different metrics to measure the prediction precision in a multifaceted way:

i) *Mean Absolute Percent Error (MAPE)* is the most commonly used metric to measure the precision for regression task. MAPE is defined as:

$$MAPE = \sum_i \frac{|\hat{y}^i - y^i|}{y^i}, \tag{18}$$

where $\hat{y}^i$ and y^i denote the predicted value and the ground-truth value, respectively.

ii) *Kendall Rank Coefficient* is defined as

$$Kendall = \frac{2(C(\hat{y}, y) - D(\hat{y}, y))}{n(n-1)}, \tag{19}$$

where $C(\hat{y}, y)$ and $D(\hat{y}, y)$ represent the number of concordant pairs and discordant pairs for $\hat{y}$ and y, respectively. This metric measures the rank similarity between predicted popularity values and the ground-truth labels.

iii) *Coverage@k* means the precision for the top k popular sources, i.e., the ratio of the sources with top k predicted popularity that are truly the top k popular sources. This metric counts the detection accuracy for some extreme popular items. We consider $k = 20$ and $k = 50$ in our experiments.

4.1.3 Comparative Methods. We select two state-of-the-art methods to compete with PreNets. One is the *Hybrid* model proposed by [22], which combines Hawkes process and feature driven perspectives. The other is the *LARM* [20], using features to separate different sequences and then adopting regression method to conduct prediction.

Moreover, in order to verify the advantages of adversarial training, we also independently train the feature driven model and the point process model to fit the data by minimizing mean square error

$$\frac{1}{B}\sum_{i=1}^{B}(h(\cdot) - y^i)^2. \tag{20}$$

The $h(\cdot)$ in (20) is implemented in distinct way for each model. For feature driven model, it is implemented by a 3-layer ELU neural network with feature vector X^i as input. For MLE-based point process model, it is implemented by LSTM with estimated intensity sequence $\mathcal{A}^i$ as input. For MLE-free point process model, it is implemented by LSTM with observed event sequence $\mathcal{S}^i$ as input.

In the following parts, we use *FD*, *PP-MB*, *PP-MF* to represent independently trained feature driven model, MLE-based and MLE-free point process models, respectively. Our MLE-based PreNets and MLE-free PreNets are respectively short as PreNets-MB and PreNets-MF.

4.2 Information Cascade on Twitter

Fig. 3 shows the learning curves for the point process model in PreNets-MB and PreNets-MF at adversarial training stage and the horizontal lines denote the MAPE for PP-MB and PP-MF. (Here we only show the MAPE result, and other metrics exhibit similar trends.) The curves show that as the training epoch increases, the MAPE for point process models decreases. During the adversarial training stage, the feedback signal from the 'critic' does help the point process models to promote the precision. Besides, comparing with the two pictures, we can find that two MLE-free models (PP-MF and PreNets-MF) performs better than two MLE-based models (PP-MB and PreNets-MB) in early-stage prediction, and the order reverses when it comes to prediction with long observation. One possible reason is that the estimated intensity sequences can be a good representation of the event sequences, but MLE on early-stage insufficient data would lead to inaccuracy of estimation.

Table 1 and Table 2 list the experiment results of several baselines and PreNets in prediction with short and long observation, respectively. Based on the results, we can summarize three-fold findings. i) MLE-free models are better than MLE-based models whether with adversarial training or not in early-stage prediction, and the opposite is true if the observation time becomes long enough.

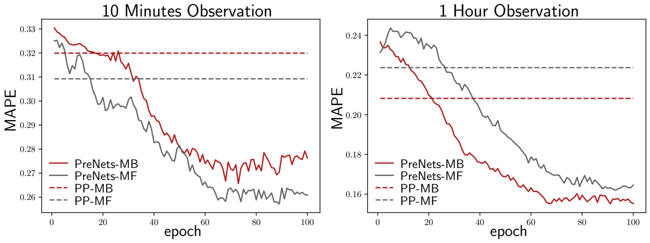

Figure 3: Learning curves for MAPE.

Table 1: 10 minutes observation for Twitter.

	MAPE	Kendall	C@50[1]	C@20
LARM	0.3088	**0.7994**	0.7807	**0.7884**
Hybrid	**0.2822**	0.7709	**0.7992**	0.7812
PP-MB	0.316	0.7586	0.6882	0.6949
PP-MF	0.2980	0.7821	0.7215	0.7556
FD	0.2862	0.7825	0.7467	0.7652
PreNets-MB	0.2811	0.7947	0.8180	0.8265
PreNets-MF	**0.2663**	**0.8147**	**0.8411**	**0.8365**

[1] The notion C@k is short for Coverage@k.

Table 2: 1 hour observation for Twitter.

	MAPE	Kendall	C@50	C@20
LARM	0.2569	0.8230	0.8659	0.8838
Hybrid	**0.1996**	0.7854	**0.9068**	0.8968
PP-MB	0.2082	**0.8270**	0.8929	**0.9029**
PP-MF	0.2237	0.8134	0.8764	0.8679
FD	0.2310	0.7922	0.8441	0.8630
PreNets-MB	**0.1574**	**0.8683**	**0.9124**	**0.9306**
PreNets-MF	0.1690	0.8321	0.8964	0.9177

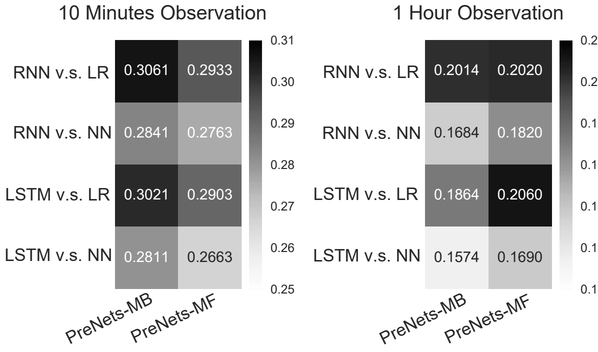

Figure 4: MAPE with different implementations for feature driven and point process models.

ii) Feature driven model performs better than point process models in early-stage prediction, while point process models achieve better precision with long observation. iii) Hybrid, PreNets-MB and PreNets-MF, as three models combining feature driven and point process perspectives achieve superior performance over other methods. For prediction with 10 minutes and 1 hour observation, our models PreNets-MB/PreNets-MF outperform all other methods, respectively achieving 0.4%/5.6% and 21.1% /15.3% improvement on MAPE.

Fig. 4 compares different model implementations for d_w, g_θ and g'_α. We compare 1-layer full-connected neural network (LR) and 3-layer ELU neural network (NN) for feature driven model d_w. Besides, we compare RNN and LSTM for point process models g_θ (MLE-based) and g'_α (MLE-free). The notion 'RNN v.s. LR' denotes that we implement point process model by LR and feature driven model by RNN. Different heat colors are used to provide a visual comparison. As we can see, more complex architectures can provide better performance. Also, changing LR to NN for feature driven model would give rise to more improvement than changing RNN to LSTM for point process models. The possible reason is that NN with more capacity could exploit more information from the features and provide better guidance to the 'interpreter'. The similar argument is given in [3].

We also apply PreNets to different observation time and study its performance variation. Fig. 5 shows the distributions of Absolute Percent Error (APE) and indicates that i) increasing observation time can reduce the median APE and also its variance, and ii) the MLE-based PreNets relies more on observation time than the MLE-free counterpart.

4.3 Movie Review on Amazon

We also run our model to predict the movie review count. At the outset, we list the basic results in Table 3 and Table 4. The experiment results are similar to those for Twitter, but have the following differences. First, the prediction performance of each method for Amazon is not as good as Twitter. We conjecture that the relatively weak correlation between the historical behaviors and the future makes it more difficult to only using temporal information to predict the future trend. Second, the LARM achieves much better precision (relative to other methods) for Amazon than Twitter. This indicates that the regression based LARM is more competent to tackle the event sequences with long life durations.

Fig. 6 shows the APE distributions for different observation time. As is depicted, the observation time plays a more important role in Amazon data set, and a small increase of the observation time would bring a big promotion of prediction precision. In comparison with Hybrid, LARM and PreNets-MB, PreNets-MF is less sensitive to observation time, which demonstrates its superiority for early-stage prediction.

Since the Amazon data set also contains the review content and user information, we extract some other features based on this information to compare the performance with only using temporal information. Event features include: the length of each review, the 'helpfulness' score of each review, the summary of each review (embedded to vectors), the history review number and averaged score of each user. Source features include: the avg. and std. of review length, the avg. and std. of 'helpfulness' scores, the sum and mean of review summary vectors, the number of influential users (whose history review number exceeds a threshold). For MLE-based point process model, event features are concatenated to the intensity sequence $\mathcal{A}^i$ as the input of g_θ. For MLE-free PreNets, we add these features to the event sequence $\mathcal{S}^i$ and input them into g'_α. For feature driven model, source features are incorporated into feature vector X^i and further input into d_w. Fig. 7 presents the experiment results, which suggest that i) other features from the perspective

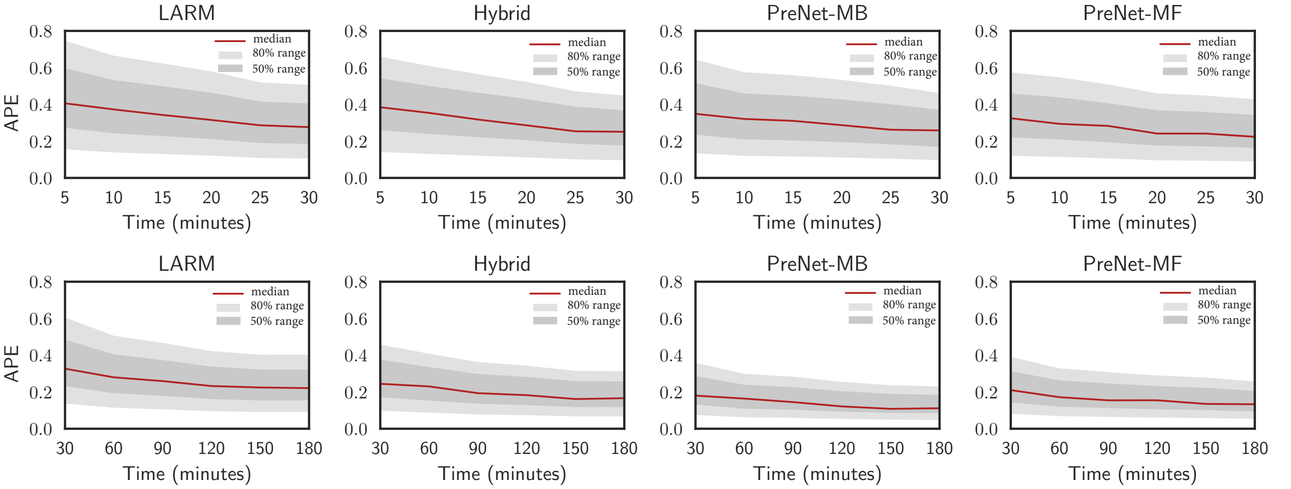

Figure 5: APE distributions for Twitter under different observation time. The red line denotes the median APE, while the light and dark gray regions respectively stand for 50th and 80th percentiles of the APE distribution.

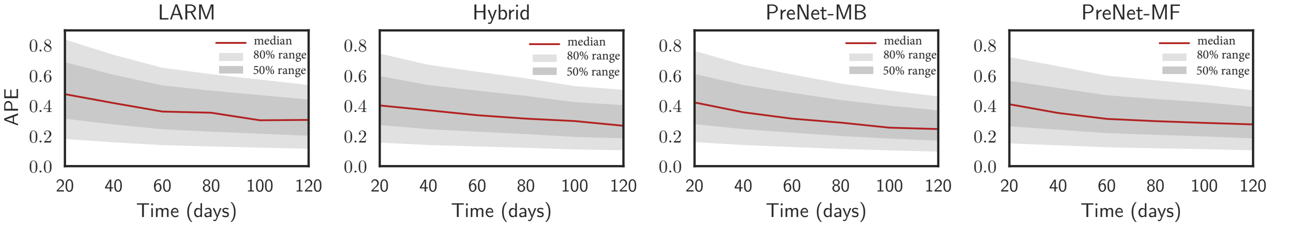

Figure 6: APE distributions for Amazon under different observation time.

Table 3: 20 days observation for Amazon.

	MAPE	Kendall	C@50	C@20
LARM	**0.3605**	**0.6786**	0.7321	0.7214
Hybrid	0.3781	0.6447	0.7341	0.7015
PP-MB	0.4244	0.6184	0.6458	0.6546
PP-MF	0.3661	0.6762	**0.7409**	**0.7285**
FD	0.4660	0.6025	0.6229	0.6591
PreNets-MB	0.3791	0.7104	0.6812	0.6790
PreNets-MF	**0.3347**	**0.7307**	**0.7577**	**0.7562**

Table 4: 100 days observation for Amazon.

	MAPE	Kendall	C@50	C@20
LARM	**0.2707**	**0.7633**	0.8356	**0.8270**
Hybrid	0.2770	0.7502	**0.8364**	0.8198
PP-MB	0.3185	0.7388	0.8236	0.8106
PP-MF	0.3323	0.6924	0.8096	0.7934
FD	0.3213	0.7209	0.8291	0.8027
PreNets-MB	**0.2536**	**0.7845**	**0.8582**	**0.8435**
PreNets-MF	0.2762	0.7719	0.8357	0.8288

of event and user information can improve the precision to a great extent, and ii) the improvement brought to early-stage prediction (about 24.3%) is much more significant than that to prediction with long observation (about 16.1%).

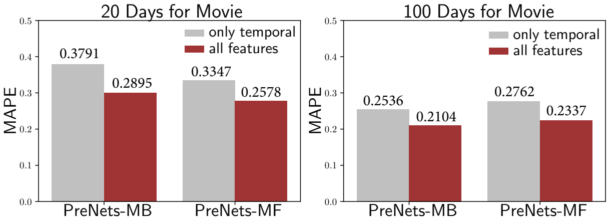

Figure 7: MAPE comparison for Amazon with only temporal features and all features.

To sum up, we summarize our empirical findings as follows. i) The adversarial training can enhance the prediction precision compared with independent training for feature driven or point process model. ii) Whether with adversarial training or independent training, the MLE-based point process model performs better than the MLE-free counterpart when the prediction is based on long observation, and the order reverses for prediction with short observation. iii) The more complex implementations for point process model and especially feature driven model can improve the precision. iv) More early features like user history records and event information can bring about precision improvement, particularly for early-stage prediction. v) Compared with the baselines, our model PreNets achieves better precision as well as less dependency on observation time and specific prediction tasks.

5 CONCLUSION

This paper proposes an adversarial training model PreNets, unifying two thinking paradigms for event popularity prediction. The framework aims at searching for an equilibrium between feature driven model, which plays as a 'critic' discriminating the predicted value from the real one, and point process model, which plays as an 'interpreter' generating a convincing predicted popularity based on the input of an observed sequence. Through a Wasserstein learning based minimax game, the precision of the 'interpreter' is enhanced with the guidance of the 'critic', leading the framework to absorb the advantages of two models. The empirical results on two different practical prediction tasks demonstrate the effectiveness and superiority of PreNets.

For future work, we will extend PreNets to other prediction tasks, such as contagion of epidemic diseases. Furthermore, as we mainly leverage temporal information to conduct prediction for generality, one possible future direction is to incorporate spatial information from social networks and delve into the spatio-temporal evolution of event sequence in the networks.

6 ACKNOWLEDGEMENTS

This work is supported by the China 973 project (2014CB340303), the National Natural Science Foundation of China (61872238, 6147-2252, 61672353), the Shanghai Science and Technology Fund (1751-0740200), the CCF-Tencent Open Research Fund (RAGR20170114), and Huawei Innovation Research Program (HO2018085286).

REFERENCES

[1] Martín Abadi and David G. Andersen. 2016. Learning to Protect Communications with Adversarial Neural Cryptography. *CoRR* abs/1610.06918 (2016).

[2] Martín Arjovsky and Léon Bottou. 2017. Towards Principled Methods for Training Generative Adversarial Networks. *CoRR* abs/1701.04862 (2017).

[3] Martín Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein Generative Adversarial Networks. In *ICML*. 214–223.

[4] Peng Bao, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2015. Modeling and Predicting Popularity Dynamics of Microblogs using Self-Excited Hawkes Processes. In *WWW*. 9–10.

[5] Justin Cheng, Lada A. Adamic, P. Alex Dow, Jon M. Kleinberg, and Jure Leskovec. 2014. Can cascades be predicted?. In *WWW*. 925–936.

[6] Rodrigo Augusto da Silva Alves, Renato Martins Assunção, and Pedro Olmo Stancioli Vaz de Melo. 2016. Burstiness Scale: A Parsimonious Model for Characterizing Random Series of Events. In *SIGKDD*. 1405–1414.

[7] D.J. Daley and David Vere-Jones. 2007. An Introduction to the Theory of Point Processes. *Springer* (2007).

[8] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. 2016. Recurrent Marked Temporal Point Processes: Embedding Event History to Vector. In *SIGKDD*. 1555–1564.

[9] Mehrdad Farajtabar, Yichen Wang, Manuel Gomez-Rodriguez, Shuang Li, Hongyuan Zha, and Le Song. 2015. COEVOLVE: A Joint Point Process Model for Information Diffusion and Network Co-evolution. In *NIPS*. 1954–1962.

[10] Shuai Gao, Jun Ma, and Zhumin Chen. 2014. Effective and effortless features for popularity prediction in microblogging network. In *WWW*. 269–270.

[11] Manuel Gomez-Rodriguez, Jure Leskovec, and Bernhard Schölkopf. 2013. Modeling Information Propagation with Survival Theory. In *ICML*. 666–674.

[12] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *NIPS*. 2672–2680.

[13] Alex Graves, Marcus Liwicki, S. Fernandez, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. 2009. A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 5 (2009), 855–868.

[14] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. 2017. Improved Training of Wasserstein GANs. In *NIPS*. 5769–5779.

[15] Hideaki Kim, Noriko Takaya, and Hiroshi Sawada. 2014. Tracking Temporal Dynamics of Purchase Decisions via Hierarchical Time-Rescaling Model. In *CIKM*. 1389–1398.

[16] Shoubin Kong, Qiaozhu Mei, Ling Feng, Fei Ye, and Zhe Zhao. 2014. Predicting bursts and popularity of hashtags in real-time. In *SIGIR*. 927–930.

[17] Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial Learning for Neural Dialogue Generation. In *EMNLP*. 2157–2169.

[18] Sha Li, Xiaofeng Gao, Weiming Bao, and Guihai Chen. 2017. FM-Hawkes: A Hawkes Process Based Approach for Modeling Online Activity Correlations. In *CIKM*. 1119–1128.

[19] Xiao Lin, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2017. Learning and Transferring Social and Item Visibilities for Personalized Recommendation. In *CIKM*. 337–346.

[20] Changsha Ma, Zhisheng Yan, and Chang Wen Chen. 2017. LARM: A Lifetime Aware Regression Model for Predicting YouTube Video Popularity. In *CIKM*. 467–476.

[21] Xiao Ma, Xiaofeng Gao, and Guihai Chen. 2017. BEEP: A Bayesian Perspective Early Stage Event Prediction Model for Online Social Networks. In *ICDM*. 973–978.

[22] Swapnil Mishra, Marian-Andrei Rizoiu, and Lexing Xie. 2016. Feature Driven and Point Process Approaches for Popularity Prediction. In *CIKM*. 1069–1078.

[23] Yosihiko Ogata. 1981. On Lewis' simulation method for point processes. *IEEE Trans. Information Theory* 27, 1 (1981), 23–30.

[24] Sasa Petrovic, Miles Osborne, and Victor Lavrenko. 2011. RT to Win! Predicting Message Propagation in Twitter. In *ICWSM*.

[25] Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. Generative Adversarial Text to Image Synthesis. In *ICML*. 1060–1069.

[26] Marian-Andrei Rizoiu, Young Lee, Swapnil Mishra, and Lexing Xie. 2017. A Tutorial on Hawkes Processes for Events in Social Media. *CoRR* abs/1708.06401 (2017).

[27] Marian-Andrei Rizoiu, Swapnil Mishra, Quyu Kong, Mark James Carman, and Lexing Xie. 2018. SIR-Hawkes: Linking Epidemic Models and Hawkes Processes to Model Diffusions in Finite Populations. In *WWW*. 419–428.

[28] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved Techniques for Training GANs. In *NIPS*. 2226–2234.

[29] Bidisha Samanta, Abir De, Abhijnan Chakraborty, and Niloy Ganguly. 2017. LMPP: A Large Margin Point Process Combining Reinforcement and Competition for Modeling Hashtag Popularity. In *IJCAI*. 2679–2685.

[30] Hua-Wei Shen, Dashun Wang, Chaoming Song, and Albert-László Barabási. 2014. Modeling and Predicting Popularity Dynamics via Reinforced Poisson Processes. In *AAAI*. 291–297.

[31] Benjamin Shulman, Amit Sharma, and Dan Cosley. 2016. Predictability of Popularity: Gaps between Prediction and Understanding. In *ICWSM*. 348–357.

[32] Bongwon Suh, Lichan Hong, Peter Pirolli, and Ed H. Chi. 2010. Want to be Retweeted? Large Scale Analytics on Factors Impacting Retweet in Twitter Network. In *IEEE SocialCom*. 177–184.

[33] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *NIPS*. 3104–3112.

[34] David Vallet, Shlomo Berkovsky, Sebastien Ardon, Anirban Mahanti, and Mohamed Ali Kâafar. 2015. Characterizing and Predicting Viral-and-Popular Video Content. In *CIKM*. 1591–1600.

[35] Duy Quang Vu, Arthur U. Asuncion, David R. Hunter, and Padhraic Smyth. 2011. Dynamic Egocentric Models for Citation Networks. In *ICML*. 857–864.

[36] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models. In *SIGIR*. 515–524.

[37] Pengfei Wang, Yanjie Fu, Guannan Liu, Wenqing Hu, and Charu C. Aggarwal. 2017. Human Mobility Synchronization and Trip Purpose Detection with Mixture of Hawkes Processes. In *KDD*. 495–503.

[38] Qun Wu, Tian Wang, Yiqiao Cai, Hui Tian, and Yonghong Chen. 2017. Rumor restraining based on propagation prediction with limited observations in large-scale social networks. In *ACSW*. 1:1–1:8.

[39] Shuai Xiao, Mehrdad Farajtabar, Xiaojing Ye, Junchi Yan, Xiaokang Yang, Le Song, and Hongyuan Zha. 2017. Wasserstein Learning of Deep Generative Point Process Models. In *NIPS*. 3250–3259.

[40] Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen M. Chu. 2017. Modeling the Intensity Function of Point Process Via Recurrent Neural Networks. In *AAAI*. 1597–1603.

[41] Jiang Yang and Scott Counts. 2010. Predicting the Speed, Scale, and Range of Information Diffusion in Twitter. In *ICWSM*.

[42] Linyun Yu, Peng Cui, Fei Wang, Chaoming Song, and Shiqiang Yang. 2015. From Micro to Macro: Uncovering and Predicting Information Cascading Process with Behavioral Dynamics. In *ICDM*. 559–568.

[43] Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. 2015. SEISMIC: A Self-Exciting Point Process Model for Predicting Tweet Popularity. In *SIGKDD*. 1513–1522.