A Variational Point Process Model for Social Event Sequences

Zhen Pan,[1] **Zhenya Huang,**[1] **Defu Lian,**[1] **Enhong Chen**[1]*
[1]Anhui Province Key Laboratory of Big Data Analysis and Application, School of Computer Science and Technology, University of Science and Technology of China
{pzhen, huangzhy}@mail.ustc.edu.cn, {liandefu, cheneh}@ustc.edu.cn

Abstract

Many events occur in real-world and social networks. Events are related to the past and there are patterns in the evolution of event sequences. Understanding the patterns can help us better predict the type and arriving time of the next event. In the literature, both feature-based approaches and generative approaches are utilized to model the event sequence. Feature-based approaches extract a variety of features, and train a regression or classification model to make a prediction. Yet, their performance is dependent on the experience-based feature exaction. Generative approaches usually assume the evolution of events follow a stochastic point process (e.g., Poisson process or its complexer variants). However, the true distribution of events is never known and the performance depends on the design of stochastic process in practice. To solve the above challenges, in this paper, we present a novel probabilistic generative model for event sequences. The model is termed Variational Event Point Process (VEPP). Our model introduces variational auto-encoder to event sequence modeling that can better use the latent information and capture the distribution over inter-arrival time and types of event sequences. Experiments on real-world datasets prove effectiveness of our proposed model.

Introduction

Events happen in real-world and on social networks. In online shopping, an event can represent user behaviors, such as click, cart or purchase (Liu et al. 2015). In geophysics, an event can be an earthquake (Sakaki, Okazaki, and Matsuo 2010). In online social media, events can be user actions (e.g., like, comment and retweet) over time, which have some features like user influence, content, time and connectivity of the social network (Fu 2011; Rizoiu et al. 2017; Shi et al. 2017). Online events usually follow the hot topics that caused by some significant news. For example, Figure 1 shows the interest about "Apple" over time in the last year[1]. Apparently, the peaks are related to special events of Apple company (e.g., product launch conference) or holidays like Christmas.

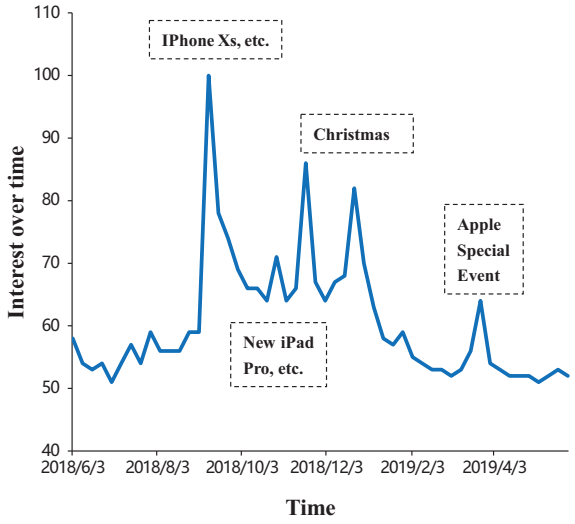


Figure 1: Google trends about "Apple". We add the relative events on the figure.

The so-called event sequences contain a series of events of different types in the continuous time domain. In an event sequence, the past events and the next event are related (Chang et al. 2015). Take the previous three types of events as examples: buyers' historical behaviors can be considered as their decision making processes. The aftershocks can happen with a month, or within days, from the main shock. Retweets could be grouped with topics of interest and timeline. The pattern of events may help cause or prevent future events. Thus, techniques to discover patterns among events are urgently required, so that the future of an event sequence can be accurately predicted (Xu et al. 2018).

In the literature, feature based methods extract relevant features and apply different machine learning algorithms to predict the type and arrival time of the future events (Naveed et al. 2011; Cheng et al. 2014; Bakshy et al. 2011; Zaman et al. 2010). However, these models heavily depend on manually selected features. It is a fatal flaw as designing features requires much expertise, especially for large-scale

*Corresponding author

[1]https://trends.google.com/trends/explore?q=apple

dataset with high dimensional features, which may severely limit its application. Recently, some other prior arts based on generative approaches are proposed, (Shen et al. 2014; Cao et al. 2017; Luo et al. 2015; Lukasik et al. 2016), in which historical events are modeled to have impact on future ones. However, the generative methods depend on the design of stochastic process and the information hidden in the sequences cannot be fully leveraged.

Variational Auto-Encoder (VAE) (Kingma and Welling 2014) is a powerful class of probabilistic models and has the ability to model complex distributions. In recent years, VAE are used in time-series (Babaeizadeh et al. 2018; Denton and Fergus 2018; Hu et al. 2017; Li and Mandt 2018). These models integrate VAE with RNN/LSTM to build a bridge between high interpretability and high predictive power. Along this line, in this paper, we present a novel probabilistic generative model for event sequences which we call *V*ariational *E*vent *P*oint *P*rocess or VEPP. Firstly, we use LSTM to embed the event sequences, so the features can be automatically extracted and utilized by the powerful neural network. Secondly, our model introduces variational auto-encoder to event sequence modeling that can use the latent information and capture the distribution over event sequences. Finally, on two real-world datasets, we find that VEPP has higher log-likelihood in the mission of predicting event type and lower error in the mission of predicting time intervals. The experiments demonstrate that VEPP can model the future of event sequences.

Related Work

In this section, we briefly summarize the related work to deal with the event prediction problem as two groups, i.e., feature-based and generative approaches.

The first category is feature based methods, which first extract some relevant features, including content, user information, original posters, network structure, and temporal features (Cheng et al. 2014; Lian et al. 2015; Wang et al. 2015). Then different machine learning algorithms are applied to build a regression or classification model, such as content-based models (Naveed et al. 2011), simple regression models (Cheng et al. 2014), regression trees (Bakshy et al. 2011) and probabilistic collaborative filtering (Zaman et al. 2010). However, these methods require much laborious feature engineering with expertise, which is hard to design, and their performance is highly sensitive to the quality of features. Besides, such approaches also have limitation in practice because they cannot be used in real-time online settings, like real-time event detection on Twitter. Given the large amount of data being produced every second, it is practically impossible to extract all the necessary features so the application is severely limited.

The second type is generative approaches which are usually based on temporal point process, like Poisson process and its complexer variants (e.g., Reinforced Poisson Processes, Hawkes Process and Self-Correcting Process). A temporal point process can be used to capture the inter-arrival times of event sequences (Daley and Vere-Jones 2007). It directly models complicated event sequences in which historical events have influences on current and future ones. Reinforced Poisson Processes (RPP) is employed to model the phenomena in social networks (Shen et al. 2014). Hawkes process, a variant of Poisson process, has been proven to be useful for describing real-world data in social network analysis (Cao et al. 2017). Furthermore, multiply variants of Hawkes process are proposed to solve the issues of event sequences. Luo et al. (2015) proposed multi-task multi-dimensional Hawkes processes for modeling event sequences. Lukasik et al. (2016) applied Hawkes processes for rumour stance classification on Twitter. However, in practice the true distribution of events is never known and the performance depends on the design of stochastic process. Besides, these methods generally are not directly optimized for future events. They cannot fully leverage the information implied in the sequences for prediction. There still remains a gap between the interpretability and predictability.

Preliminaries

In this section, we first give the problem definition, and then briefly introduce the two basic models for the temporal point process and Variational Auto-Encoders.

Problem definition

As shown in Figure 2, the input is a sequence of events $x_{1:n} = (x_1, \cdots, x_n)$, where x_n is the n-th event. The event $x_n = (k_n, \tau_n)$ is represented by the event type $k_n \in \{1, 2, \cdots, K\}$ (K discrete event classes) and the time interval $\tau_n \in \mathbb{R}^+$. The time interval $\tau_n = t_n - t_{n-1}$ is the difference between the starting time of event x_{n-1} and x_n. Given a sequence of events $x_{1:n-1}$, the event sequence modeling task is to produce a distribution over the event type k_n and the time interval τ_n of the next happening event. We aim to develop probabilistic models to predict what and when the next event will happen.

Temporal point process

A temporal point process is a random process which is used to capture the time intervals of event sequences (Daley and Vere-Jones 2007). A temporal point process is characterized by the conditional intensity function $\lambda(t_n|x_{1:n-1})$, which is conditioned on the past events $x_{1:n-1}$. The conditional intensity is the expected infinitesimal rate at which events are expected to occur around time t. Given the $n-1$ past events, the probability density function for the time interval of next event is:

$$f(\tau_n|x_{1:n-1}) = \lambda(\tau_n|x_{1:n-1})e^{-\int_0^{\tau_n}\lambda(u|x_{1:n-1})\mathrm{d}u}. \quad (1)$$

The Poisson process (Kingman 2005) is the simplest and most ubiquitous example of point process, which assumes that events occur independently of one another. The conditional intensity is $\lambda(\tau_n|x_{1:n-1}) = \lambda$, where λ is a positive constant.

Furthermore, more complex point processes have been proposed, like Hawkes Process (Hawkes 1971) and Self-Correcting Process (Isham and Westcott 1979). All these processes try to model the dependency on the past events. For example, Hawkes process is a self-exciting process in which the arrival of an event causes the conditional intensity

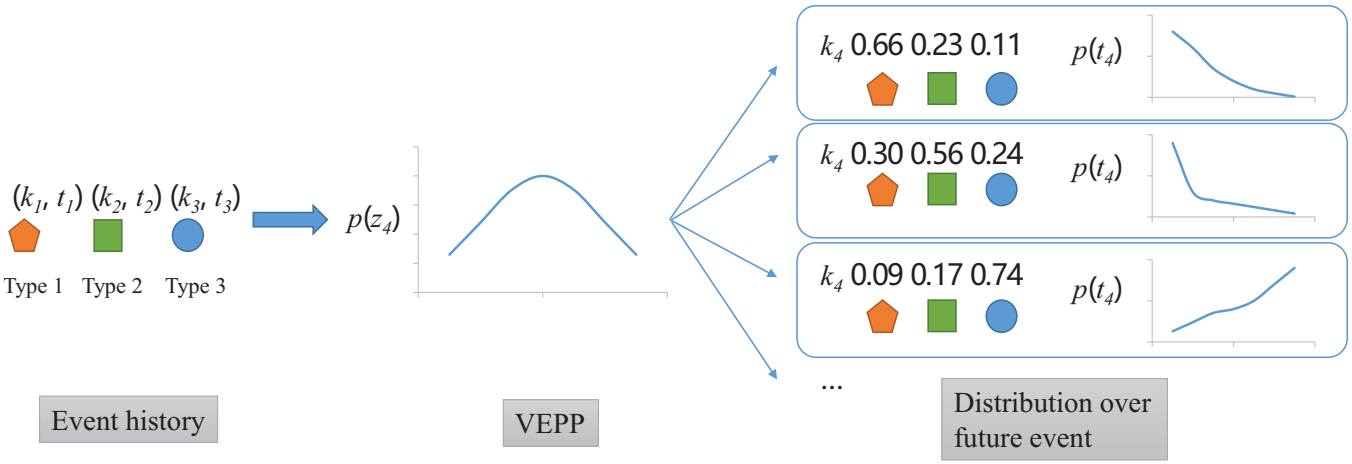


Figure 2: The data flowchart of VEPP.

function to increase. The conditional intensity of Hawkes Process is:

$$\lambda(t_n|x_{1:n-1}) = \lambda_0(t_n) + \sum_{i=1}^{n-1} \phi(t - T_i), \tag{2}$$

where $T_i < t$ are all the event time having occurred before current time t_n, and which contribute to the event intensity at time t_n. $\lambda_0(t_n)$ is a deterministic base intensity function, and ϕ is called the memory kernel.

However, the true model of the dependencies is never known in practice (Mei and Eisner 2017) and the performance depends on the design of conditional intensity. In this work, we learn a variational model that fits the conditional intensity by the history of events.

Variational Auto-Encoders

A Variational Auto-Encoder is a generative model which can effectively model complex multimodal distributions over the data space. A VAE introduces a set of latent random variables z, designed to capture the variations in the observed variables x. The joint distribution is defined as: $p_\theta(z|x) \propto p_\theta(x|z)p_\theta(z)$. The simple prior $p_\theta(z)$ is usually chosen to be a multivariate Gaussian. The parameters of complex likelihood $p_\theta(x|z)$ are produced by neural networks. Approximating the intractable posterior $p_\theta(z|x)$ with a recognition neural network $q_\phi(z|x)$ the parameters of the generative model θ as well as the recognition model ϕ can be jointly optimized by maximizing the evidence lower bound (ELBO) $\mathcal{L}$ on the marginal likelihood $p_\theta(x)$:

$$\begin{aligned} \log p_\theta(x) &= \mathrm{KL}(q_\theta||p_\theta) + \mathcal{L}(\theta, \phi) \\ &\geq \mathcal{L}(\theta, \phi) = -\mathbb{E}_{q_\phi}\left[\log \frac{q_{\phi(z|x)}}{p_\theta(z, x)}\right]. \end{aligned} \tag{3}$$

Recent works apply VAEs to time-series data including video (Babaeizadeh et al. 2018; Denton and Fergus 2018; Mehrasa et al. 2019), text (Hu et al. 2017), and audio (Li and Mandt 2018; Chung et al. 2015). Such models usually integrate a time-step VAE with RNN/LSTM. The ELBO thus becomes a summation of time-step-wise variational lower bound:

$$\begin{aligned} \mathcal{L}_{\theta,\phi} = \sum_{n=1}^{N} \Big[&\mathbb{E}_{q_\phi(z_{1:n}|x_{1:n})}[\log p_\theta(x_n|x_{1:n-1}, z_{1:n})] \\ &- \mathrm{KL}(q_\phi(z_n|x_{1:n})||p_\theta(z_n|x_{1:n-1}))\Big]. \end{aligned} \tag{4}$$

Variational Event Point Process

In this section, we give the details of our VEPP model. We propose a generative model for event sequence modeling by using the VAEs. Figure 3 shows the architecture of our model. Overall, the types of events and their time intervals are encoded using a recurrent VAE model. At each step, the model uses past events to create a distribution over latent codes z_n, a sample of which is then decoded into two probability distributions: one over the possible event types and another over the time intervals for the next event.

Event representing and embedding

As shown in Figure 3(a), at time step n, the model takes the event x_n as input, which is the prediction target, and also the past events $x_{1:n-1}$. These inputs are used to product a conditional distribution $q_\phi(z_n|x_{1:n})$ from which a latent code z_n is sampled. The true distribution over latent variables z_n is intractable. We rely on a time-dependent inference network $q_\phi(z_n|x_{1:n})$ that approximates it with a conditional Gaussian distribution $\mathcal{N}(\mu_{\phi_n}, \sigma^2_{\phi_n})$. To prevent z_n from just copying x_n, we force $q_\phi(z_n|x_{1:n})$ to be close to the prior distribution $p(z_n)$ using a Kullback-Leibler divergence term.

At each step during training, a latent variable z_n is drawn from the posterior distribution $q_\phi(z_n|x_{1:n})$. The output event $\hat{x}_n$ is then sampled from the distribution $p_\theta(x_n|z_n)$ of the conditional generative model which is parameterized by θ. For convenience, we assume the event type and time inter-

vals are conditionally independent given the latent code z_n:

$$p_\theta(x_n|z_n) = p_\theta(k_n, \tau_n|z_n) = p^k_\theta(k_n|z_n)p^\tau_\theta(\tau_n|z_n), \quad (5)$$

where $p^k_\theta(k_n|z_n)$ and $p^\tau_\theta(\tau_n|z_n)$ are the conditional generative model for event type and time interval, respectively. It is a standard assumption in event prediction (Du et al. 2016). The sequential model generates two probability distributions: a categorical distribution over the event types and a temporal point process over the time interval for the next event.

The event types are modeled with a multinomial distribution in which case k_n can only take a finite number of values:

$$\sum_{i=1}^{K} p^k_\theta(k_n = i|z_n) = 1, \quad (6)$$

where $p^k_\theta(k_n = i|z_n)$ is the probability that event type i will occur, and K is the total number of event types.

The time interval follows an exponential distribution whose parameter is $\lambda(z_n)$, similar to a standard temporal point process model:

$$p^\tau_\theta(\tau_n|z_n) = \lambda(z_n)e^{-\lambda(z_n)\tau^n} \quad \text{if } \tau_n \geq 0, \quad (7)$$

where $p_\theta(\tau_n|z_n)$ is a probability density function over variable τ_n and $\lambda(z_n)$ is the intensity of the temporal point process, which depends on the latent variable sample z_n.

At step n, the current event x_n is represented as a vector x^{emb}_n with a two-step embedding strategy. First, we compute a representation for the event type k_n and the time interval τ_n separately. Then, we concatenate these two representations and get a new representation x^{emb}_n of the event:

$$\begin{aligned} k^{emb}_n &= Emb_k(k_n), \\ \tau^{emb}_n &= Emb_\tau(\tau_n), \\ x^{emb}_n &= Emb_{k,\tau}([k^{emb}_n, \tau^{emb}_n]). \end{aligned} \quad (8)$$

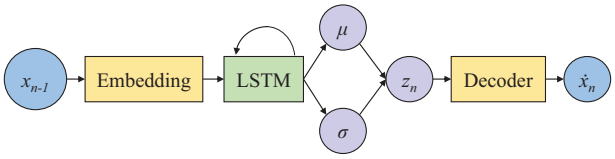
Here, Emb_k, Emb_τ and $Emb_{k,\tau}$ represent the embedding functions. A one-hot encoding is used to represent the event type k_n.

Generation

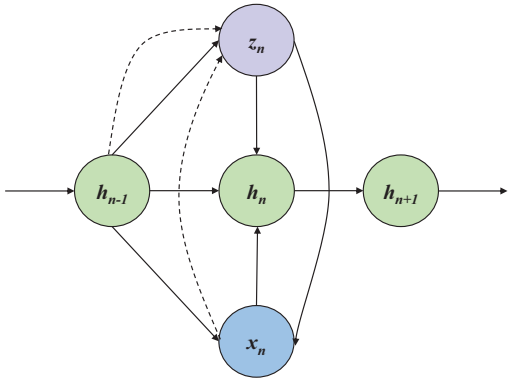
The VEPP contains a VAE at every time step. However, these VAEs are conditioned on the state variable h_{n-1} of an LSTM. It will help the VAE to take into account the temporal structure of the sequential data. Unlike a standard VAE, the prior on the latent random variable is no longer a standard Gaussian distribution, but follows the distribution:

$$\begin{aligned} z_n &\sim \mathcal{N}(\mu_n, \sigma^2_n), \\ \mu_n, \sigma^2_n &= f^{prior}(h_{n-1}), \end{aligned} \quad (9)$$

where μ_n and σ_n are the parameters of the conditional prior distribution and f^{prior} can be any highly flexible function such as neural networks.



(a) At each time step, the model uses the history of event sequences and inter-arrival times to generate a distribution over latent codes.



(b) Graphical illustrations of operations of the VEPP.

Figure 3: Architecture of VEPP.

Firstly, we sample z_n from the prior to generate an event at step n. The parameters of the prior distribution are calculated based on the past $n-1$ events $x_{1:n-1}$. Then, an event type $\hat{k}_n$ and time interval $\hat{\tau}_n$ are generated as follows:

$$\begin{aligned} \hat{k}_n &\sim p^k_\theta(k_n|z_n), \\ \hat{\tau}_n &\sim p^\tau_\theta(\tau_n|z_n). \end{aligned} \quad (10)$$

The decoder network for event type $f^k_\theta(z_n)$ is a MLP with a softmax output to generate the probability distribution in Equation (6):

$$p^k_\theta(k_n|z_n) = f^k_\theta(z_n). \quad (11)$$

The decoder network for time interval $f^\tau_\theta(z_n)$ is another MLP, producing the parameter of the point process model for temporal distribution in Equation (7):

$$\lambda(z_n) = f^\tau_\theta(z_n). \quad (12)$$

The LSTM encodes the current event and the past events into a vector representation:

$$h_n = LSTM_\phi(x^{emb}_n, z^{emb}_n, h_{n-1}). \quad (13)$$

Recurrent networks turn variable length sequences into meaningful, fixed-sized representations. The parameterization of the generative model results in the factorization:

$$p(x_{1:N}, z_{1:N}) = \prod_{n=1}^{N} p(x_n|z_{1:n}, x_{1,n-1})p(z_n|x_{1:n-1}, z_{1:n-1}) \quad (14)$$

Inference

The posterior is proportional to the product of the likelihood and the prior. So the approximate posterior will not only be

Table 1: Statistics of datasets

Dataset	K	number of sequences		number of event tokens		sequence length		
		train	test	train	test	min	mean	max
Retweets	3	20,000	2,000	2,156,116	216,405	50	109	264
MemeTrack	4,895	96,391	2,470	383,548	10,440	2	5	31

a function of x_n but also of h_{n-1} following the equation:

$$z_n|x_n \sim \mathcal{N}(\mu_{z,n}, \sigma^2_{z,n}),$$
$$\mu_{z,n}, \sigma^2_{z,n} = Enc(x_n, h_{n-1}), \quad (15)$$

where $\mu_{z,n}$ and $\sigma_{z,n}$ denote the parameters of the approximate posterior. The encoding of the approximate posterior and the decoding for generation are tied through the LSTM hidden state h_{n-1}. This conditioning on h_{n-1} results in the factorization:

$$q(z_{1:N}, x_{1:N}) = \prod_{n=1}^{N} q(z_n|x_{1:n}, z_{1:n-1}). \quad (16)$$

Learning

We train the model by optimizing the variational lower bound over the entire sequence comprised of N steps:

$$\mathcal{L}_{\theta,\phi}(x_{1:N}) = \sum_{n=1}^{N} (\mathbb{E}_{q_\phi(z_n|x_{1:n})}[\log p_\theta(x_n|z_n)] - \mathrm{KL}(q_\phi(z_n|x_{1:n})||p_\theta(z_n|x_{1:n-1}))). \quad (17)$$

Given the latent code z_n, the event type and time interval are conditionally independent, so the log-likelihood can be written as follows:

$$\mathbb{E}_{q_\phi(z_n|x_{1:n})}[\log p_\theta(x_n|z_n)] = \mathbb{E}_{q_\phi(z_n|x_{1:n})}[\log p^k_\theta(k_n|z_n)] + \mathbb{E}_{q_\phi(z_n|x_{1:n})}[\log p^\tau_\theta(\tau_n|z_n)]. \quad (18)$$

Given the form of p^k_θ, the log-likelihood reduces to a cross entropy between the predicted event type $p^k_\theta(k_n|z_n)$ and the ground truth k^*_n. Given the ground truth time interval τ^*_n, we calculate its log-likelihood over a small time interval Δ_τ under the predicted distribution.

$$\log \left[\int_{\tau^*_n}^{\tau^*_n + \Delta_\tau} p^\tau_\theta(\tau_n|z_n) \mathrm{d}\tau_n \right] = \log(1 - e^{-\lambda(z_n)\delta_t}) - \lambda(z_n)\tau^*_n. \quad (19)$$

Experiments

In this section, we evaluate the performance of VEPP on two real-world datasets, i.e., Retweets Dataset (Zhao et al. 2015) and MemeTrack Dataset (Leskovec, Backstrom, and Kleinberg 2009).

Datasets

Retweets dataset The Retweets dataset includes $166,076$ retweet sequences, each corresponding to some original tweet. Each retweet event is labeled with the retweet time relative to the original tweet creation, so that the time of the original tweet is 0. Each retweet event is also marked with the number of followers of the retweeter. As usual, we assume that these $166,076$ streams are drawn independently from the same process, so that retweets in different streams do not affect one another.

Unfortunately, the dataset does not specify the identity of each retweeter, only his or her popularity. To distinguish different kinds of events that might have different rates and different influences on the future, following previous study of Mei and Eisner (2017), we divide the events into $K = 3$ types: retweets by "small", "medium" and "large" users. Small users have fewer than 120 followers (50% of events), medium users have fewer than $1,363$ (45% of events), and the rest are large users (5% of events). Given the past retweet history, our model must learn to predict how soon it will be retweeted again and how popular the retweeter is (i.e., which of the three types).

We randomly sampled disjoint train and test sets with $20,000$ and $2,000$ sequences respectively. We truncated sequences to a maximum length of 264, which affected 20% of them. For computing training and test likelihoods, we treated each sequence as the complete set of events observed on the interval $[0, T]$, where 0 denotes the time of the original tweet, which is not included in the sequence, and T denotes the time of the last tweet in the truncated sequence.

MemeTrack dataset The MemeTrack Dataset considers the reuse of fixed phrases, or "memes", in online media. It contains time-stamped instances of meme use in articles and posts from 1.5 million different blogs and news sites, spanning 10 months from August 2008 till May 2009, with several hundred million documents.

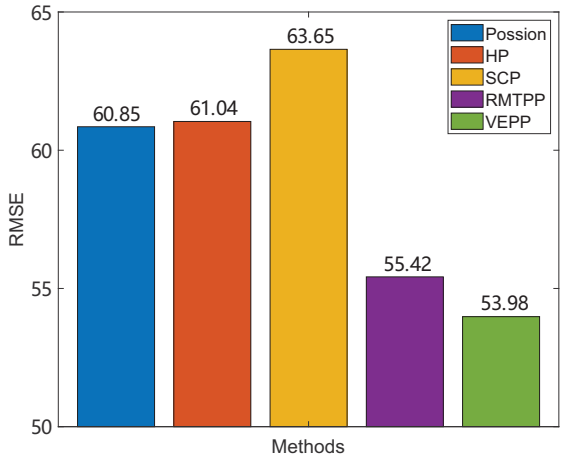
The K event types correspond to the different websites. Given one meme's past trajectory across websites, our model can learn to predict how soon and where it will be mentioned again.

We followed the previous study of Gomez-Rodriguez et al. (2013) to process the dataset, which selected the top $5,000$ websites in terms of the number of memes they mentioned. We truncated sequences to a maximum length of 31 and selected the minimum length of 2. We randomly sampled disjoint train and test sets with $96,391$, and $2,470$ sequences respectively, treating them as before.

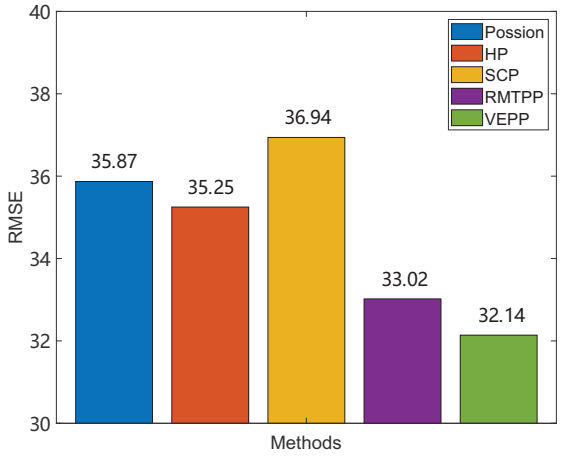
Table 1 shows statistics of the two datasets. The two datasets have very different characteristics.

Architecture details

The VEPP model architecture is shown in Figure 3. Event type and inter-arrival time inputs are each passed through 2-layer MLPs with ReLU activation. They are then concate-



(a) RMSE on Retweets dataset.



(b) RMSE on MemeTrack dataset.

Figure 4: Comparison of RMSE on Retweets and MemeTrack dataset.

nated and followed with a linear layer. Numbers of hidden nodes of LSTMs for Retweets and MemeTrack datasets are 256 and 64, respectively. Networks are 2-layer MLPs, with ReLU activation after the first layer. Dimension of the latent code is 256. Event decoder is a 3-layer MLP with ReLU at the first two layers and softmax for the last one. The time decoder is also a 3-layer MLP with ReLU at the first two layers, with an exponential non-linearity applied to the output to ensure the parameter of the point process is positive.

Implementation details

The models are implemented with TensorFlow (Abadi et al. 2016) and are trained using the Adam (Kingma and Ba 2015) optimizer for $1,000$ epochs with batch size 32 and learning rate 0.001. We split both datasets into training and test sets containing 70% and 30% of samples respectively. We select the best model during training based on the model loss (18) on the test set.

Baselines

Poisson Process The intensity function is a constant, which produces an estimate of the average inter-event gaps.

Hawkes Process (HP) Hawkes process is a self-exciting point process, in which past events from the history conspire to raise the intensity of each type of events. Such excitation is positive, additive over the past events, and exponentially decaying with time.

Self-Correcting Process (SCP) We fit a self-correcting process with the intensity function in the book of Daley and Vere-Jones (2007).

Recurrent Marked Temporal Point Processes (RMTPP) RMTPP is proposed in the study of Du et al. (2016). It views the intensity function of a temporal point process as a non-linear function of the history, and uses a recurrent neural network to automatically learn a representation of influences from the event history. Compared with VEPP, RMTPP does not have the stochastic latent code that models diverse distributions over event type and time interval.

Table 2: Log-likelihood of event type prediction

Dataset	Model	Log-likelihood	Error (%)
Retweets	HP	-7.06	49.48
	RMTPP	-6.88	38.02
	VEPP	$\mathbf{-6.48}$	$\mathbf{37.68}$
MemeTrack	HP	-802.2	90.37
	RMTPP	-14.3	86.85
	VEPP	$\mathbf{-10.7}$	$\mathbf{85.04}$

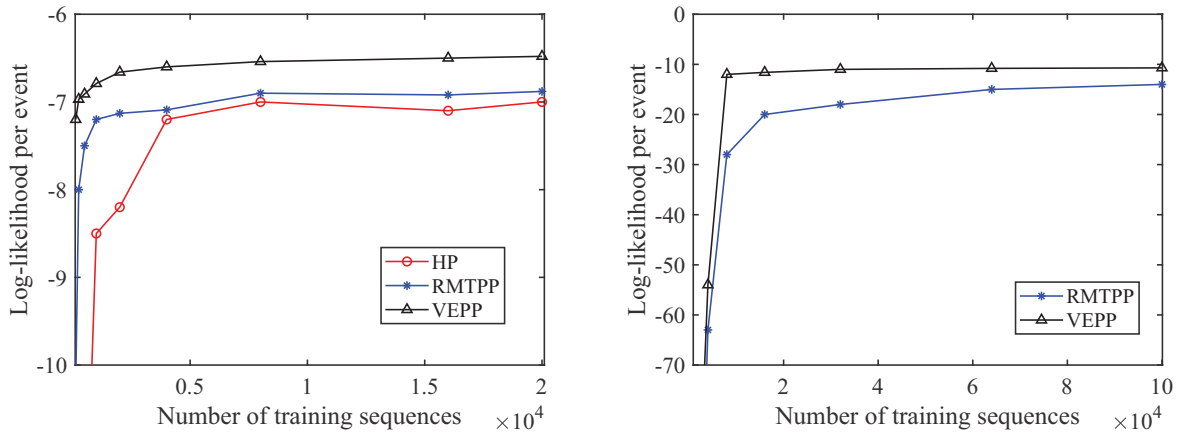
Metrics

We use log-likelihood (LL) of event type to compare our model with the HP and RMTPP. For Poisson Process and Self-Correcting Process, their performance on event type prediction are similar to Hawkes Process and not very satisfactory.

$$\begin{aligned} \text{LL} &= \log \prod_{i=1}^{n} f(x_i|\Omega) \\ &= \sum_{i=1}^{n} \log f(x_i|\Omega) = \sum_{i=1}^{n} l(\Omega|x_i). \end{aligned}$$

We also compare Root Mean Square Error (RMSE) of inter-arrival time prediction.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{\tau}_i - \tau_i)^2}$$



(a) Learning curve on Retweets dataset.

(b) Learning curve on MemeTrack dataset.

Figure 5: Log-likelihood of HP, RMTPP and VEPP on Retweets and MemeTrack dataset.

Table 3: Log-likelihood for VEPP with different latent variable dimensionality

Latent size	32	64	128	256	512
Retweets	-7.08	-6.58	-6.58	-6.48	-6.68
MemeTrack	-10.9	-10.7	-10.5	-10.2	-10.0

Experiment Results

Overall performance Table 2 shows experimental results that compare VEPP with HP and RMTPP. VEPP outperforms HP and RMTPP on both Retweets and MemeTrack datasets. We believe that this is because the VEPP model is better in modeling the complex distribution over future events. For Retweets dataset, three methods all have relatively good performance. The reason may be the events of Retweets dataset have less types, so the patterns of event sequences are easy to model. Correspondingly, for MemeTrack dataset, VEPP and RMTPP significantly outperform HP. It is may because the event types are nearly $5,000$ and much larger than Retweets datasets. It also proves that VAE can use the latent information of event sequences and VEPP has the ability to modeling the complex distribution over future events. The prediction error is high for MemeTrack dataset due to the large number of types.

Figure 4 shows RMSE in predicting the time interval given the history of previous events. VEPP achieves the lowest error, i.e., outperforms the other methods under the metric. The three methods based on point process have relatively higher error, because the designed point process does not fit the real situation. RMTPP and VEPP achieve better results, since they can learn the complex distribution. While VEPP can also use the latent information over the event sequence, it performs even better.

Learning curves We compare the learning curves of HP, RMTPP and VEPP in terms of the number of train sequences, as shown in Figure 5. In Figure 5(b), HP preforms not well and cannot be displayed on the figure because of the figure size. It shows how the performance changes when training data are increased. We can see that our VEPP model outperforms RMTPP, and both significantly outperform the Hawkes process. These observations demonstrate the robustness and effectiveness of VEPP model in event prediction.

Sensitive of latent variable dimensionality of LSTM We next explore the architecture of our model by varying the sizes of the latent variable. Table 3 shows the log-likelihood of our model for different sizes of the latent variable. We see that as we increase the size of the latent variable, we can model a more complex latent distribution which results in better performance.

Conclusion

We presented a novel probabilistic model for sequence data, a variational auto-encoder that captures uncertainty in event types and arrival time. As a generative model, it could produce event sequences by sampling from a prior distribution, the parameters of which were updated based on neural networks that control the distributions over the next event type and temporal occurrence. The model could also be used to analyze given input sequences of events to determine the likelihood of observing particular sequences. We demonstrated empirically that the model is effective for capturing the uncertainty inherent in event prediction. In future, we will take into account the structural and contextual information to model the event sequences.

Acknowledgments

This research was partially supported by grants from the National Key Research and Development Program of China (No. 2016YFB1000904), and the National Natural Science Foundation of China (Grants No. U1605251, 61727809).

References

Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation*.

Babaeizadeh, M.; Finn, C.; Erhan, D.; Campbell, R. H.; and Levine, S. 2018. Stochastic variational video prediction. In *6th International Conference on Learning Representations*.

Bakshy, E.; Hofman, J. M.; Mason, W. A.; and Watts, D. J. 2011. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*.

Cao, Q.; Shen, H.; Cen, K.; Ouyang, W.; and Cheng, X. 2017. Deephawkes: bridging the gap between prediction and understanding of information cascades. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*.

Chang, B.; Zhu, F.; Chen, E.; and Liu, Q. 2015. Information source detection via maximum a posteriori estimation. In *2015 IEEE International Conference on Data Mining*.

Cheng, J.; Adamic, L.; Dow, P. A.; Kleinberg, J. M.; and Leskovec, J. 2014. Can cascades be predicted? In *Proceedings of the 23rd international conference on World wide web*.

Chung, J.; Kastner, K.; Dinh, L.; Goel, K.; Courville, A. C.; and Bengio, Y. 2015. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*.

Daley, D. J., and Vere-Jones, D. 2007. *An introduction to the theory of point processes: volume II: general theory and structure*. Springer Science & Business Media.

Denton, E., and Fergus, R. 2018. Stochastic video generation with a learned prior. In *Proceedings of the 35th International Conference on Machine Learning*.

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 *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Fu, T.-c. 2011. A review on time series data mining. *Engineering Applications of Artificial Intelligence*.

Gomez-Rodriguez, M.; Leskovec, J.; and Schölkopf, B. 2013. Modeling information propagation with survival theory. In *Proceedings of the 30th International Conference on Machine Learning*.

Hawkes, A. G. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*.

Hu, Z.; Yang, Z.; Liang, X.; Salakhutdinov, R.; and Xing, E. P. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*.

Isham, V., and Westcott, M. 1979. A self-correcting point process. *Stochastic Processes and Their Applications*.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*.

Kingma, D. P., and Welling, M. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations*.

Kingman, J. F. C. 2005. Poisson processes. *Encyclopedia of biostatistics*.

Leskovec, J.; Backstrom, L.; and Kleinberg, J. 2009. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Li, Y., and Mandt, S. 2018. Disentangled sequential autoencoder. In *Proceedings of the 35th International Conference on Machine Learning*.

Lian, D.; Xie, X.; Zheng, V. W.; Yuan, N. J.; Zhang, F.; and Chen, E. 2015. Cepr: A collaborative exploration and periodically returning model for location prediction. *ACM TIST*.

Liu, Q.; Zeng, X.; Zhu, H.; Chen, E.; Xiong, H.; Xie, X.; et al. 2015. Mining indecisiveness in customer behaviors. In *2015 IEEE International Conference on Data Mining*.

Lukasik, M.; Srijith, P.; Vu, D.; Bontcheva, K.; Zubiaga, A.; and Cohn, T. 2016. Hawkes processes for continuous time sequence classification: an application to rumour stance classification in twitter. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

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 *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Mehrasa, N.; Jyothi, A. A.; Durand, T.; He, J.; Sigal, L.; and Mori, G. 2019. A variational auto-encoder model for stochastic point processes. In *IEEE/CVF Computer Vision and Pattern Recognition*.

Mei, H., and Eisner, J. M. 2017. The neural hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*.

Naveed, N.; Gottron, T.; Kunegis, J.; and Alhadi, A. C. 2011. Bad news travel fast: A content-based analysis of interestingness on twitter. In *Proceedings of the 3rd international web science conference*.

Rizoiu, M.-A.; Lee, Y.; Mishra, S.; and Xie, L. 2017. A tutorial on hawkes processes for events in social media. *arXiv preprint arXiv:1708.06401*.

Sakaki, T.; Okazaki, M.; and Matsuo, Y. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*.

Shen, H.; Wang, D.; Song, C.; and Barabási, A.-L. 2014. Modeling and predicting popularity dynamics via reinforced poisson processes. In *Twenty-eighth AAAI conference on artificial intelligence*.

Shi, L.-l.; Liu, L.; Wu, Y.; Jiang, L.; and Hardy, J. 2017. Event detection and user interest discovering in social media data streams. *IEEE Access*.

Wang, Y.; Yuan, N. J.; Lian, D.; Xu, L.; Xie, X.; Chen, E.; and Rui, Y. 2015. Regularity and conformity: Location prediction using heterogeneous mobility data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Xu, T.; Zhu, H.; Zhong, H.; Liu, G.; Xiong, H.; and Chen, E. 2018. Exploiting the dynamic mutual influence for predicting social event participation. *IEEE Transactions on Knowledge and Data Engineering*.

Zaman, T. R.; Herbrich, R.; Van Gael, J.; and Stern, D. 2010. Predicting information spreading in twitter. In *Workshop on computational social science and the wisdom of crowds, nips*.

Zhao, Q.; Erdogdu, M. A.; He, H. Y.; Rajaraman, A.; and Leskovec, J. 2015. Seismic: A self-exciting point process model for predicting tweet popularity. In *KDD 2015*.