Neural Stochastic Differential Equations with Bayesian Jumps for Marked Temporal Point Process

Kazi Islam
Department of Computer Science
University of California, Riverside
Riverside, CA 92507
kazi.islam@email.ucr.edu

Christian Shelton
Department of Computer Science
University of California, Riverside
Riverside, CA 92507
cshelton@cs.ucr.edu

Abstract

Many real-world systems evolve according to continuous dynamics and get interrupted by stochastic events i.e. systems that both flow (often described by a differential equation) and jump. If the equation of the continuous motion is unknown, the stochastic event generation process can be modeled as samples generated from a *marked temporal point process*, in the form of event sequences with *non-uniform* time intervals. Additionally, each event is marked with the type of the event and a real-valued *noisy* measurement. The noisy measurements themselves are the realizations of a stochastic process with an unknown stochastic differential equation (SDE). We present a framework that simultaneously models the intensity function of the temporal point process and learn stochastic process governing the distribution of the observed noisy measurements. Similar to the underlying system of interest, the latent dynamics of our framework evolves continuously according to ordinary differential equations (ODEs) while, the jumps at the observations are controlled by Gated Recurrent Units (GRUs) through a *Bayesian* update, which accounts for the observation noise. We present preliminary data fitting results on a real-world medical dataset.

1 Introduction and Related Work

The real world has many examples of systems that evolve continuously in time and are interrupted by stochastic events, spaced by non-uniform time intervals. In many domains, such as social networks, global politics, and computer systems, the "equations of motion" are unknown and the unknown dynamics of the underlying system can only be estimated by records of the events and their real-valued times. Data in the medical domain, for example, electronic health records (EHRs), can additionally contain real-valued *noisy* physiological measurements of different variables including vital signs, drugs, and lab tests. These observations are inherently noisy due to charting errors, calibration, or device noise. Our goal is to model the dynamics of such systems where in addition to the real-valued irregularly sampled times, each event is accompanied by the type of the event and also a real-valued noisy measurement.

Given such event sequences with non-uniform time intervals, marked temporal point process and intensity functions are a powerful mathematical framework to model the event generation process. The measurement variables can be modeled by the Fokker-Plank equation [1] and the measurements viewed as noisy observations of this underlying stochastic process. Recurrent Neural Networks (RNNs) are not a natural fit for modeling such data, as they naturally assume the observations are regularly spaced across the temporal dimension. Usual tricks to adapt RNNs for irregular time series data divide the timeline into equally-spaced time-intervals [2–4], use the time difference between the observations as an input to the model [5, 6], or decay the hidden state exponentially when no

Preprint. Under review.

observation is made [5,7]. However, these approaches either make the simplified “missing at random” assumption, fail to capture the continuously evolving dynamics of the underlying system or use a fixed form of evolution in between observations.

We present a framework based on the recently proposed neural ordinary differential equations (Neural ODEs) [8] for jointly learning the two intertwined stochastic processes, that is the temporal point process generating the stochastic events and the stochastic process generating the associated real-valued noisy measurements. We encode the state of the underlying system in a latent state vector $h(t)$, which evolves piecewise-continuously according to a neural ODE. The jumps at the observation times are controlled by GRUs [9], which changes the trajectory of $h(t)$. Additionally, a Bayesian update term accounts for the noise in the measurements to learn the true distribution of the underlying multivariate stochastic process of the measurement variables. The intensity function of the point process and the mark distributions are generated from $h(t)$ with additional neural nets.

In a neural ordinary differential equation (Neural ODE) framework, the differential equation expressing the flow dynamics is parameterized by a neural network without considering potential jumps of the latent state. Jia et al. [10] address the jumps using a neural net and also an adjoint-operator formulation of the gradient for learning in constant memory and model the intensity of point processes and distribution of real-valued features. However, they ignore the stochasticity of the underlying process generating the noisy measurements. Brouwer et al. [11] proposes a Bayesian update network using GRUs to regularize the neural ODE to better track a true belief state for the underlying system without directly addressing the event generation process concerning event history. Rubanova et al. [12] proposes an ODE-RNN hybrid where the hidden state dynamics flow according to a neural ODE and the jumps according to a GRU unit. They applied their model for learning the intensity function without accounting for the event history and did not address the noise in the measured observations. We address all these issues in modeling noisy, irregularly-sample time series altogether using a unified framework combining the above ideas.

Finally, we present preliminary data fitting results on a real-world electronic health records from the MIMIC-III (’Medical Information Mart for Intensive Care’) database [13], a publicly available critical care database.

2 Preliminaries

We want to model the dynamics of a hybrid system whose samples contain a finite set of events of unbounded random size; each event has a real-valued time and a mark consisting of a discrete-valued label and a real-valued noisy measurement of that label. Formally, each sample $S = \{(t_1, k_1), (t_2, k_2), \ldots, (t_I, k_I)\}$ is an event sequence with non-uniform time intervals, where t_i is the time and k_i is the mark (the label of the event and the corresponding recorded noisy measurement) associated with event i and $t_i < t_{i+1}$. We let I represent the total number of events and T be the total length of time ($T \geq t_I$).

In many real-world systems, measurements of multiple types of labels are recorded at the same time (for instance, vital signs like heart rate, blood pressure, and respiratory rate of a patient). Therefore, we extend the mark $\boldsymbol{k_i}$ to consist of a pair of vectors, $(\boldsymbol{e_i}, \boldsymbol{v_i})$, where $\boldsymbol{e_i} \in \{0,1\}^{|L|}$ is a multi-hot encoding of the set of all L types of event labels (heart rate, glucose, blood pH, etc.), and $\boldsymbol{v_i} \in \mathbb{R}^{|L|}$ is a vector of values with one element for each of the L types of event labels. Event labels that co-occur have 1s in the multi-hot encoding representation and the corresponding elements in $\boldsymbol{v_i}$ are the associated noisy measurements. Elements of the vector $\boldsymbol{v_i}$ corresponding to the event-types that did not occur at time t_i are represented with 0s.

We postulate that the measurements $\boldsymbol{v_i}$ are observations of an L-dimensional continuous stochastic process $\mathcal{V}(t)$, which is driven by an unknown stochastic differential equation (SDE):

$$d\mathcal{V}(t) = M(\mathcal{V}(t))dt + \Sigma(\mathcal{V}(t))dW(t), \tag{1}$$

where $dW(t)$ is a Wiener process. The distribution of $\mathcal{V}(t)$ evolves according to the Fokker-Planck equation. The continuous mean and the covariance function for the probability density function (PDF) of $\mathcal{V}(t)$ is denoted by $M(t)$ and $\Sigma(t)$ in equation 1. From the noisy measurements observed at irregular times, we need to model the two unknown mean and the covariance function. Note that, observations from all dimensions are not sampled at the same time, which results in the missing values

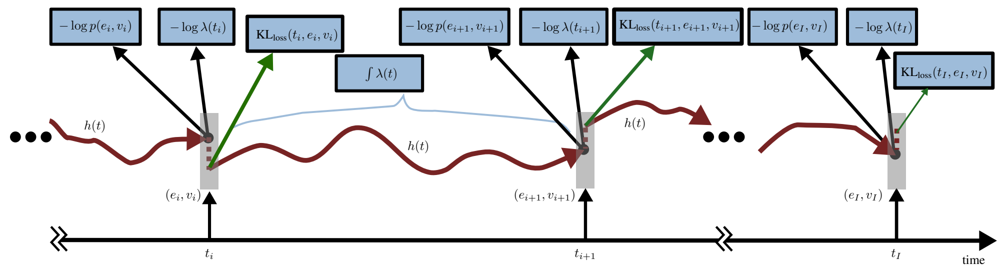

Figure 1: Illustration of our proposed framework. Just before processing input observation (e_i, v_i), the negative log-likelihood loss for the intensity and the mark distribution is calculated from the continuous hidden state, obtained from the solution of the ODE solver at t_i. After the GRU (shown by the gray box) processes the observation, the hidden state jumps to the new state (indicated by the brown dots) and the KL divergence loss (shown by the green arrows) is computed (equation 13).

in v_i. Additionally, we assume the observation noise in the sampled measurement for dimension l is drawn from a Gaussian distribution with zero mean and standard deviation of σ_l^{obs}.

We postulate that the observation times are drawn from a general continuous-time point process, whose intensity may depend on the history of both $\mathcal{V}$ and the previous observation times. Its conditional intensity function, $\lambda(t, \mathcal{H})$, may depend on both the raw time t, and the history up through time t, $\mathcal{H}$. If we let $\mathcal{H}_i$ be the observed history up to but not including event i, $\mathcal{H}_i = \{(t, k) \mid (t, k) \in S \wedge t < t_i\}$, the density of the distribution over the time of event i, given the history is:

$$P(t_i \mid \mathcal{H}_i) = \lambda(t_i, \mathcal{H}_i) e^{-\int_{t_{i-1}}^{t_i} \lambda(s, \mathcal{H}_i)\, ds} \; . \tag{2}$$

For a marked point process observed within a time interval $[0, T)$, the number of observed events I is finite, and the continuous log-likelihood of such a finite event-stream is given by,

$$\sum_{i=1}^{I} \left[\log P(t_i \mid \mathcal{H}_i) + \log P(k_i \mid \mathcal{H}_i, t_i)\right] + \log P(t_{I+1} > T \mid \mathcal{H}) \; . \tag{3}$$

The middle $P(k_i \mid \mathcal{H}_i, t_i)$ term is a combination of the distribution for the event label, $\boldsymbol{e_i}$, and the complex marginal distribution over $\mathcal{V}(t_i)$ conditioned on $\mathcal{H}_i$, as implied by the Fokker-Planck equation. We assume, the distribution for the event label, e_i, is drawn from its own distribution conditioned on history: $P(\boldsymbol{e_i} \mid \mathcal{H}_i, t_i)$. As the labels are discrete-valued, this distribution could be folded in to the intensity, to create a separate intensity for each event label possibility (with the total intensity being the sum of the intensities over all possible labels). However, for our model, a global intensity and a multinomial distribution over labels is more natural.

If the SDE (Equation 1) were known, the inference would depend on tracking the belief state of $\mathcal{V}$: $P(\mathcal{V}(t) \mid \mathcal{H})$. Rather than parameterizing and estimating the SDE and then developing an (almost assuredly approximate) belief-state filtering method for $\mathcal{V}$, we propose to learn the belief-state filter directly, never explicitly modeling the underlying SDE. To this end, we next describe a continuous-time neural ODE process with jumps that we train to track both the SDE and the point process. Thus, its hidden state, h, serves as a sufficient summary of the history to model the belief state of $\mathcal{V}$ and the history, $\mathcal{H}$, as necessary for the conditional intensity, $\lambda(t, \mathcal{H})$.

3 Proposed Framework

Figure 1 schematically shows the components of our framework. We describe each component in the sections below.

3.1 ODE-GRU

Recently proposed Neural ODEs [8] are a family of continuous-time models where the hidden state $h(t)$ is defined as a solution to an ODE initial-value problem (IVP):

$$\dot{h(t)} = f_\theta(h(t), t), \tag{4}$$

where $h(t_0) = h_0$ and $f_\theta(h(t), t)$ is a multi-layer perception (MLP) with parameters θ.

The continuous hidden state $h(t)$ can be evaluated at any set of times using a numerical ODE solver:

$$h_0, ..., h_I = ODESolve(f_\theta, h_0, (t_0, ..., t_I)). \tag{5}$$

Similar to the ODE-RNN model [12], the hidden state in our framework evolves according to an ODE in between the noisy observations, and the solution to the ODE at time t is the input to an RNN unit that provides the jump at observations. For the RNN unit, we choose the Gated Recurrent Unit (GRU) [9]. This jump allows the hidden state to incorporate the observation and track the belief state of the underlying system. Formally,

$$h_i = ODESolve(f_\theta, h'_{i-1}, (t_{i-1}, t_i)), \tag{6}$$

where $h'_i = GRU(h_i, k_i)$. That is, h_i is the hidden state before the observation at time t_i, and h'_i is the hidden state after the observation at time t_i.

The hidden state is coupled with the processes through four MLPs, g^λ, g^e, g^μ, and g^{var}:

$$\lambda(t) = g^\lambda(h(t)) \tag{7}$$
$$p(\boldsymbol{e_i} \mid t_i, h(t_i)) = \mathcal{B}\left(\boldsymbol{e_i}; g^e(h(t_i))\right) \tag{8}$$
$$p(\boldsymbol{v_i} \mid t_i, h(t_i)) = \mathcal{N}\left(\boldsymbol{v_i}; g^\mu(h(t_i)), g^{\text{var}}(h(t_i))\right) \tag{9}$$

where g^λ maps h to the intensity of an observation event (a non-negative scalar), g^e maps h to $|L|$ independent Bernoulli distributions (that is, the $|L|$ independent probabilities that each label appears at an observation), g^μ maps h to $|L|$ means, and g^{var} maps h to $|L|$ variances. For $\boldsymbol{v_i}$, we assume that, conditioned on the hidden state h, each observed label's value is drawn independently from a normal distribution. Thus, $g^e_l(h_i)$ is the model's predicted probability that label l appears in the observation at time t_i. Similarly, $g^\mu_l(h_i)$ and $g^{\text{var}}_l(h_i)$ are the mean and variance of the predicted normal distribution over the value associated with label i, if it appears in the ith observation.

3.2 Maximizing the Log-likelihood

We train the model using maximum likelihood estimation. The log-likelihood of the event stream S over the finite time interval T is derived from equation 2 and 3 as:

$$\log(p(S|\mathcal{W})) = \sum_i [\log \lambda(t_i \mid h(t_i); \mathcal{W}) + \log p(k_i \mid h(t_i), t_i; \mathcal{W})] - \int_{t=0}^{T} \lambda(t)\, dt, \tag{10}$$

where $\mathcal{W}$ is the set of all parameters including the ODE-GRU parameters and the MLP parameters controlling the dynamics, such as the event intensities, and the joint density of the events and their values. The $-\int_{t=0}^{T} \lambda(t)\, dt$ term integrates the log-probability of the infinitely many non-events that did not occur within the the time interval T.

The model evolves according to equation 6 where h_i is equivalent to $h(t_i)$, the history, $\mathcal{H}_i$, prior to event i. The intensity for event i and the mark distribution $p(k_i|h(t_i), t_i)$ in equation 10 therefore is obtained from h_i, which is the solution of the ODE-solver with initial value h'_{i-1} (state after the GRU processes event $i-1$), at time t_{i-1}.

3.3 Bayesian Jump

We add the Bayesian Jump framework [11] to regularize the system by encouraging its output to track the Bayes filter and account for the noise in the observed measurements. Consider the observation of the lth label's value at time t_i. Just before the observation event, the model's belief over its value has a mean and variance of $g^\mu_l(h_i)$ and $g^{\text{var}}_l(h_i)$. After making the observation, the hidden state jumps to h'_i. The new belief has a mean and variance of $g^\mu_l(h'_i)$ and $g^{\text{var}}_l(h'_i)$. We would like these beliefs to be consistent with a Bayesian update of a belief state, assuming a known zero-mean observation noise with variance σ^2_{obs}.

Assuming we observe value $\boldsymbol{v_{i,l}}$, the posterior distribution over the value of event type l should have mean and variance

$$\mu_{\text{Bayes},l} = \frac{\sigma_{\text{obs}}^2 \cdot g_l^{\mu}(h_i)}{g_l^{\text{var}}(h_i) + \sigma_{\text{obs}}^2} + \frac{g_l^{\text{var}}(h_i) \cdot \boldsymbol{v_{i,l}}}{g_l^{\text{var}}(h_i) + \sigma_{\text{obs}}^2} \tag{11}$$

$$\sigma_{\text{Bayes},l}^2 = \frac{g_l^{\text{var}}(h_i) \cdot \sigma_{\text{obs}}^2}{g_l^{\text{var}}(h_i) + \sigma_{\text{obs}}^2} \,. \tag{12}$$

If we let $p_{\text{Bayes},l}$ be this Bayesian update normal distribution (with mean $\mu_{\text{Bayes},l}$ and variance $\sigma_{\text{Bayes},l}^2$) and let $p'_{i,l}$ be the normal distribution predicted by the model *after* the GRU update (with mean $g^{\mu}(h'_i)$ and variance $g^{\text{var}}(h'_i)$), we add a KL-divergence loss to encourage the two distributions to be the same:

$$\text{KL}_{\text{Loss}}(t_i, \boldsymbol{e_i}, \boldsymbol{v_i}) = \sum_l \boldsymbol{e_{i,l}} D_{\text{KL}}(p_{\text{Bayes},l} \parallel p'_{i,l}) \,, \tag{13}$$

which sums over the individual event type observations, because the model assumes that each component is independent given h.

The joint loss to optimize then becomes:

$$-\log(p(S|\mathcal{W})) + \sum_i \text{KL}_{\text{Loss}}(t_i, \boldsymbol{e_i}, \boldsymbol{v_i}) \,, \tag{14}$$

which can be optimized using gradient descent.

To aid the network in making the correct update, we augment its input. For the i^{th} event, instead of directly feeding the mark k_i to the GRU, $\boldsymbol{v}_i$ is appended with the means and variances of the predicted outputs ($g_l^{\mu}(h_i)$ and $g_l^{\text{var}}(h_i)$ for all l) and the normalized error terms $(\boldsymbol{v}_{i,l} - g_l^{\mu}(h_i))/\sqrt{g_l^{\text{var}}(h_i)}$ (for all l), which are then multiplied by a dimension specific weight vector W_l and passed through a ReLU non-linearity to produce $\boldsymbol{q_i}$. Non-observed dimensions in $\boldsymbol{q}_i$ is zeroed out and is appended with $\boldsymbol{e}_i$ to be passed as input to the GRU. While this extra information is already available in h_i, the prevents the GRU from needing to learn to recalculate it.

4 Experiments

One of the prominent use-cases of irregular time series modeling is patient time series data observed in an Intensive Care Unit (ICU) setting, which are highly noisy, sparse, and irregularly sampled. We perform our experiments on a subset of the publicly available MIMIC-III database curated as a benchmark for evaluating machine learning models in healthcare settings [3], containing 17 physiological time-series variables. Unlike the proposed methods in the corresponding paper [3], we do not discretize the data, instead, take into account each time where at least one variable is measured, and represent the measurement variables and their values as e_i and v_i (section: 2).

We fit our models on the first 24 hours of data on a training set containing 5000 samples. We report the data-fitting performance on a separate hold-out set with 1000 samples. To properly use batching and achieve speedup in training, we rounded the observation times into the nearest minute and took the aggregate of measurements taking place in the same rounded minute. This results in $(24 * 60 + 1)$ or 1441 possible measurement times. Note that, rounding to the minute results in very little information loss compared to hourly aggregation. All the ODEs in a minibatch are solved continuously for each 1441 measurement times, however, the jumps using the RNN occur only at the observation times for each sample. A separate boolean mask matrix indicates the times where an observation occurred for each sample in the minibatch.

We used ODE Solvers from torchdiffeq python package [8], particularly the fifth-order "dopri5" solver with adaptive step size. The relative and absolute tolerances were set as 1e-3 and 1e-4 respectively. The adjoint method described in [8] and [10] can be used to reduce the memory use, with the cost of added computational time. We used Adam optimizer [14] for learning the parameters of all the MLPs and the RNN. The integral term in equation 10 can be directly obtained by augmenting the ODE with an integral over $\lambda(t)$, which is obtained from the hidden state using g^{λ}.

We refer to our model as ODE-RMTPP and compare the performance between the two versions of it: with and without the Bayesian Jump framework. We report the total negative log-likelihood (NLLH) loss and also the individual losses: the intensity loss, event loss, and the value loss.

Table 1: MIMIC-III Results

Model	Sequence NLLH	Intensity NLLH	Event NLLH	Value NLLH
ODE-RMTPP	390424.3438	75319.7109	146406.2812	146339.5938
ODE-RMTPP + BAYES	**383080.7500**	**75302.3359**	**145825.2656**	**139586.7969**

Results (Table: 1) show that adding the Bayes filter improves the data-fitting performance, as the NLLH in the test set reduces for each distribution. The most significant reduction in the NLLH loss occurs for the value distribution. This makes sense, as the Bayes Filter is particularly tracking the belief state for the values. However, the KL divergence loss also implicitly acts as a regularizer, which results in loss reduction for both the intensity and the event distribution.

5 Discussion and Future Work

We have developed a framework based on Neural Ordinary Differential Equations, with a Bayesian Jump Framework to simultaneously capture the irregularity and noise in time-series data. Our model is best suited for temporal point process modeling in time-series data where, the irregularly sampled temporal event-sequences are marked with a real-valued noisy measurement, in addition to the type of the event e.g. medical time series. Currently, in our framework, only the jump of the hidden state at the observation times is Bayesian. One of the possible future directions is to make the network fully Bayesian by imposing priors on the parameters of the MLPs and the RNN.

References

[1] H. Risken and H. Haken, *The Fokker-Planck Equation: Methods of Solution and Applications Second Edition*. Springer, 1989.

[2] Z. C. Lipton, D. C. Kale, and R. Wetzel, "Modeling missing data in clinical time series with rnns," *Machine Learning for Healthcare*, 2016.

[3] H. Harutyunyan, H. Khachatrian, D. C. Kale, and A. Galstyan, "Multitask learning and benchmarking with clinical time series data," *arXiv preprint arXiv:1703.07771*, 2017.

[4] H. Suresh, N. Hunt, A. E. W. Johnson, L. A. Celi, P. Szolovits, and M. Ghassemi, "Clinical intervention prediction and understanding using deep networks," *CoRR*, vol. abs/1705.08498, 2017.

[5] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *arXiv preprint arXiv:1606.01865*, 2016.

[6] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, "Recurrent marked temporal point processes: Embedding event history to vector," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1555–1564, ACM, 2016.

[7] H. Mei and J. Eisner, "The neural Hawkes process: A neurally self-modulating multivariate point process," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (USA), pp. 6757–6767, Curran Associates Inc., 2017.

[8] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *NeurIPS*, 2018.

[9] K. Cho, B. van Merrienboer, Çaglar Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *ArXiv*, vol. abs/1406.1078, 2014.

[10] J. Jia and A. R. Benson, "Neural jump stochastic differential equations," *CoRR*, vol. abs/1905.10403, 2019.

[11] E. D. Brouwer, J. Simm, A. Arany, and Y. Moreau, "Gru-ode-bayes: Continuous modeling of sporadically-observed time series," *CoRR*, vol. abs/1905.12374, 2019.

[12] Y. Rubanova, R. T. Q. Chen, and D. Duvenaud, "Latent odes for irregularly-sampled time series.," *CoRR*, vol. abs/1907.03907, 2019.

[13] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database," *Scientific data*, vol. 3, 2016.

[14] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.