Point Processes for Competing Observations with Recurrent Networks (POPCORN): A Generative Model of EHR Data

Shreyas Bhave SAB2323@CUMC.COLUMBIA.EDU
Department of Biomedical Informatics
Columbia University
New York, NY, USA

Adler Perotte AJP2120@CUMC.COLUMBIA.EDU
Department of Biomedical Informatics
Columbia University
New York, NY, USA

Abstract

Modeling EHR data is of significant interest in a broad range of applications including prediction of future conditions or building latent representations of patient history. This can be challenging because EHR data is multivariate and irregularly sampled. Traditional treatments of EHR data involve handling irregular sampling by imputation or discretization. In this work, we model the full longitudinal history of a patient using a generative multivariate point process that simultaneously: (1) Models irregularly sampled events probabilistically without discretization or interpolation (2) Has a closed-form likelihood, making training straightforward (3) Encodes dependence between times and events with an approach inspired by competing risk models (4) Allows for direct sampling. We show improved performance on next-event prediction compared to existing approaches. Our proposed framework could potentially be used in many different contexts including prediction, generation of synthetic data and building latent representations of patient history.

1. Introduction

Multivariate, irregularly sampled time series data are ubiquitous in many data modalities across healthcare, including principally Electronic Health Records (EHR) data. They are defined in the context where a dataset contains a set of time series where each time series contains a sequence of pairs $\{(t_i, e_i)\}_{i=1}^{N}$ where t_i represents the time and e_i represents a particular event type. In many cases, past sequences strongly inform which events are likely to happen in the future and when. In the case of EHRs, each time series is a longitudinal history of a patient's visits, lab tests, administration of medications, diagnoses of conditions and more. Modeling EHRs as sequences of such events and building better generative models is of interest in a wide range of applications including prediction of future events (e.g. conditions, readmission), building latent representations of a patient's history and generation of synthetic data. Prior generative approaches for modeling this kind of data are lacking in one or more of the following key characteristics: (1) times and events are considered conditionally independent given history which can be limiting for prediction and simulation (2) direct sampling is not possible (3) optimization is challenging due to a lack of a closed-form likelihood. In healthcare, events and times are often tightly linked. In particular, if the next event happens within minutes versus after many days, this will change our prediction about what the next event is.

We model EHR data as a Multivariate Temporal Point Process, a probabilistic framework for modeling irregularly sampled data. In this framework, both the time until the next event and the type of event are modeled probabilistically by conditioning on a summary of the entire history prior to that point. Our main contribution is that we propose a multivariate model which simultaneously (1) specifies dependence between events and times inspired by competing risks (2) allows for direct sampling (3) specifies a closed-form likelihood, making stochastic optimization straightforward.

We evaluate the model on datasets commonly used in the point process literature: a MIMIC-II (Johnson et al., 2016) dataset consisting of ICU visits where the events are conditions and their timestamps and a Stack Overflow dataset which consists of two years of data on users receiving sequences of badges on the online forum. We further evaluate our model on synthetic EHR data from Synthea (Walonoski et al., 2018) and Synthea (Ear Infection) generated in prior related work (Enguehard et al., 2020). These datasets are publicly available, which allows for data transparency and for direct comparison to relevant prior work. The model is compared against recently proposed approaches which differ in key ways as outlined in the Related Works. We evaluate the models on both prediction of event type given next event time as well as joint probability of next event and next time on a held-out test set. The particular metrics we use to assess these are weighted F1/AUROC and negative log likelihood normalized by time respectively.

Generalizable Insights about Machine Learning in the Context of Healthcare

The majority of predictive modeling approaches built on EHR longitudinal data make simplifying assumptions either when modeling feature inputs or the output events of interest. When modeling irregularly sampled time series features, the approach is often to discretize the irregularly sampled sequence into equal bins and develop an interpolation model for data that is missing prior to using a standard approach (e.g. LSTM) for regularly sampled data (Che et al., 2018). Such an approach suffers from both loss of information and introduction of noise. The other criticism of many prediction models, and more specifically survival models, is that they do not handle competing risks. Without taking competing risks into account, model estimation and prediction can be biased due to misspecification.

In this work, we model the full longitudinal history of a patient using a multivariate point process model that has several advantages: (1) Irregularly sampled events are modeled directly without discretization or interpolation (2) A closed-form likelihood makes training straightforward (3) The model encodes dependence between times and events with an approach inspired by competing risk (4) Direct sampling is possible. We show improved performance with EHR data on next-event prediction compared to other approaches. Our results provide evidence that incorporating competing risks is important for modeling EHR data especially in the context of next-event prediction.

2. Related Work

Neural temporal point process models have garnered substantial interest in recent years with the emergence of neural density estimation approaches. These methods all employ the basic framework of a temporal point process but differ in the following key categories (1) independence assumptions between events and times (2) the probabilistic object which is modeled (e.g. conditional intensity function, cumulative intensity, conditional probability density) (3) the approach used to encode past history to predict next event (e.g. continuous LSTM, GRU, etc.). As a result of the choices made in each of these categories, models have different properties. Favorable properties as outlined in Shchur et al. (2019) include (1) a closed-form likelihood for ease of optimization (2) direct sampling (of next event and time given history) for ease of use (3) distributional flexibility.

In one of the earliest works in neural point processes, Du et al. (2016) use a simple RNN to encode history, reading in data as tuples of times and events. They use the hidden state of the RNN h_j to model the conditional intensity function which has a fixed specification. With this specification, the time until next event is a unimodal distribution. They also model the next event as conditionally independent of next time. As such, the flexibility of the model is restricted by the exponential specification and next time and event are not tightly coupled. Additionally, the history encoding approach does not directly handle irregular sampling.

The neural hawkes process (Mei and Eisner, 2016) addresses many of these issues. They specify a multivariate point process which does take competing risks into account. Additionally, they employ an approach which uses a custom continuous time LSTM architecture in an attempt to better encode history. The main drawback of this approach is that it chooses to model the conditional intensity function which reduces the efficiency of optimization by requiring a Monte Carlo estimate of an integral. Additionally, sampling requires a thinning algorithm.

Intensity-free temporal point processes (Shchur et al., 2019) take the approach of directly modeling the conditional probability of the next event time using mixture density networks, avoiding the issues that arise from modeling conditional intensities. This allows for direct sampling and a closed-form likelihood. However, they model times independently of events. Additionally, they use the same architecture as Du et al. (2016) to model history which does not account for irregular sampling.

Several other methods (Okawa et al. (2019), Omi et al. (2019), Taddy et al. (2012), Tabibian et al. (2017)) have been proposed which use different approaches to model conditional intensity functions which suffer from similar issues as those outlined above.

In our model, we attempt to integrate the most favorable properties from prior work to develop an approach which attempts to handle the primary dependencies of EHR data. Our model is a multivariate point process with dependencies between events and times, directly models the conditional probabilities of each event given history, and employs a multi-channel neural architecture to model the irregularly sampled signal for encoding history.

3. Background

3.1. Temporal Point Process

A temporal point process (TPP) is a random process which is meant to model a sequence of N times $(t_0, t_1, \ldots, t_N)$. Such a process is defined by specifying a distribution for the interevent times, or the times between successive events conditioned on history up until each successive point $\mathcal{H}_{t_{n-1}}$. A TPP is fully specified by the joint density $f(t_0, t_1, \ldots, t_N) = \prod_n f(t_n|\ldots t_{n-2}, t_{n-1}) = \prod_n f(t_n|\mathcal{H}_{t_{n-1}})$. The traditional method of modeling this data is to use a conditional intensity function $\lambda^*(t) = \lambda_\theta(t|\mathcal{H})$ where θ is the set of model parameters and the star denotes that the intensity is conditioned on all historical times. This intensity function describes the instantaneous rate at which an event happens given that the event hasn't happened yet: $\lambda^*(t) = \lim_{dt \to 0} \frac{P(t \leq T < t+dt)}{dt * S(t)} = \frac{f(t)}{S(t)}$. Reasoning about the intensity function instead of the density allows for the specification of well-established self-excitation processes, such as the Hawkes process. In the general case, with a parametric form of the intensity specified, maximum likelihood estimation is possible but can involve certain challenges. The likelihood is as follows: $\sum_{i=1}^{N} \log p_\theta^*(t_i) = \sum_{i=1}^{N} \log \lambda_\theta^*(t_i) - \int_0^{t_N} \lambda_\theta^*(s)ds$ as shown in Rasmussen (2018). The difficulty arises in choosing a flexible parametric specification for the intensity function that still has a closed form integral. Shchur et al. (2019) address this issue by directly modeling $p_\theta^*(t)$ in the setting where times and events are considered independently.

3.2. Multivariate Temporal Point Process

A multivariate temporal point process is defined as a random process that is used to model event streams. An event stream is a sequence of N events $\{(t_i, e_i)\}_{i=1}^N$ where t_i is the time that the ith event occurs and $e_i \in \mathcal{E}$ is the event type chosen from a set of possible events $\mathcal{E}$. A key characteristic of a truly multivariate point process is that the events are tightly coupled with the times. This dependence is traditionally characterized by the conditional intensity function for each event $\lambda_e^*(t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} P(t \leq T < t + \Delta t, E = e | T \geq t, \mathcal{H})$ which is also known as a cause-specific hazard function. This hazard function represents the instantaneous rate at which a given event is happening in the presence of competing events. The hazard functions for each event completely specify a joint likelihood over the entire sequence which can be derived as follows:

$$
\begin{aligned}
\lambda^*(t) &= \sum_{e=1}^{|\mathcal{E}|} \lambda_e^*(t) \\
P(T_i \leq t|\mathcal{H}) &= 1 - \exp\left(-\int_{t_{i-1}}^{t} \sum_{e=1}^{|\mathcal{E}|} \lambda_e^*(t)dt\right) \\
P(T_i = t|\mathcal{H}) &= \exp\left(-\int_{t_{i-1}}^{t} \sum_{e=1}^{|\mathcal{E}|} \lambda_e^*(t)dt\right) * \sum_{e=1}^{|\mathcal{E}|} \lambda_e^*(t) \\
P(E_i = e|T_i = t, \mathcal{H}) &= \frac{\lambda_e^*(t)}{\sum_{e=1}^{|\mathcal{E}|} \lambda_e^*(t)} \\
P(T_i = t, E_i = e|\mathcal{H}) &= \lambda_e^*(t) * \exp\left(-\int_{t_{i-1}}^{t} \sum_{e=1}^{|\mathcal{E}|} \lambda_e^*(t)dt\right)
\end{aligned}
\tag{1}
$$

Prior approaches which model the conditional intensity functions thus do incorporate competing risks of all the events but at the cost of the necessity to take a Monte Carlo estimate of the integral in the second term of the objective function. This specification also complicates the sampling process as it typically requires a thinning algorithm.

4. The POPCORN Model

4.1. Construction of Objective

In our approach, which we call **PO**int **P**rocesses for **C**ompeting **O**bservations with **R**ecurrent **N**etworks, or POPCORN, instead of modeling the conditional intensity, we directly model the conditional probability of each event given history $p_e^*(t) = p_e(t|\mathcal{H})$. We note that this is distinct from the joint probability in (1) which is often labeled in a similar way as in (Enguehard et al., 2020). Our model makes the assumption that the conditional probabilities of each of the event time distributions are conditionally independent given history.

We gain several advantages from directly modeling the conditional probabilities including the ability to directly sample and a simple, closed-form likelihood, while maintaining flexibility by using a mixture density network to model each conditional probability.

Given this, we derive our objective as follows where $p_e^*(t)$ is the conditional probability given history, $S_e^*(t)$ is the survival function given history and $h_e^*(t)$ is the hazard function given history.

$$
\begin{aligned}
P(T_i = t, E_i = e|\mathcal{H}) &= P(T_i = t|\mathcal{H})P(E = e|T_i = t, \mathcal{H}) \\
P(T_i \leq t|\mathcal{H}) &= 1 - \prod_{e=1}^{|\mathcal{E}|} S_e^*(t) \\
P(T_i = t|\mathcal{H}) &= \left(\prod_{e=1}^{|\mathcal{E}|} S_e^*(t)\right)\left(\sum_{e=1}^{|\mathcal{E}|} \frac{p_e^*(t)}{S_e^*(t)}\right) \\
&= \left(\prod_{e=1}^{|\mathcal{E}|} S_e^*(t)\right)\left(\sum_{e=1}^{|\mathcal{E}|} h_e^*(t)\right) \\
P(E_i = e|T_i = t, \mathcal{H}) &\propto p_e^*(t) \prod_{j \neq e} S_j^*(t) \\
P(T_i = t, E_i = e|\mathcal{H}) &= \frac{p_e^*(t)\prod_{j\neq e} S_j^*(t)}{\sum_{e=1}^{|\mathcal{E}|} p_e^*(t)\prod_{j\neq e} S_j^*(t)} \left(\prod_{e=1}^{|\mathcal{E}|} S_e^*(t)\right)\left(\sum_{e=1}^{|\mathcal{E}|} h_e^*(t)\right) \qquad (2)
\end{aligned}
$$

If the conditional probability and the survival function can easily be computed, this likelihood is closed-form and it is straightforward to conduct stochastic optimization. By modeling each of the conditional probabilities separately, we can sample from this model simply by taking a sample from all of the event distributions and taking the minimum time as our next time and event. This is described in more detail in the next section.

4.2. Sampling and Connection to Competing Risk

In competing risk problems, a key idea is that there are latent or potential failure times $T_1, \ldots T_e$. A multiple decrement, or joint survival function, can be described as follows where z is a feature vector and we have e different event types:

$$Q(t_1, \ldots t_e; \mathbf{z}) = P(T_1 > t_1, \ldots T_e > t_e, z) \qquad (3)$$

In this setting, the data which is observed can be described in the following way:

$$T = min\{T_1, \ldots T_e\}, E = \{j|T_j \leq T_k, k = 1 \ldots e\} \qquad (4)$$

This extends to the setting of point processes but the interpretation becomes that there is a separate competing risk problem for each timestep for a given patient. In our case, we can directly specify this joint survival function because we model the probabilities of each event separately given history and assume conditional independence. This means that the joint survival function is simply the product of each of the survival distributions of the conditional probability densities.

Thus, sampling is straightforward: (1) Sample from each of the conditional distributions to get a set of $t_1 \ldots t_{|\mathcal{E}|}$ and take the minimum. (2) This minimum provides both the time until the next event and the event itself.

4.3. Conditional Independence Assumption and Identifiability

The assumption of conditional independence may at first appear restrictive. However, as Tsiatis (1975) shows: given any joint survival function with arbitrary dependencies between events, there exists a different joint function which is specified by independent risks that models the data just as precisely. This result makes it impossible to test whether competing risks are independent. For our purposes, this theorem shows that given we have a sufficiently flexible way of modeling each conditional distribution, we should be able to recover an equivalent model to any model which incorporates dependent risks.

4.4. Mixture Density Networks and Distributional Specifications

In order to specify a flexible distributional specification for each of the conditional probabilities we choose to use mixture density networks. In particular, we use a mixture of Weibull distributions and a mixture of Fréchet distributions for all our experiments.

Mixture of Weibulls The Weibull distribution is a common distribution for specifying survival in survival analysis because its parameters have a direct interpretation. It has a shape (k) and a scale (l) parameter, where the shape parameter controls whether the hazard is increasing or decreasing overtime. Thus, a mixture of Weibulls could capture the combination of many different possible hazard shapes. We use an MLP to generate parameters for the Weibull and the mixture weights (w) from the historical encoding. A Softplus transform is used to ensure that the parameters are restricted to positive real numbers and weights are normalized. The pdf for a mixture of Weibulls is the following:

$$p(t; \boldsymbol{l}, \boldsymbol{k}, \boldsymbol{w}) = \sum_{i=1}^{J} w_i \frac{k_i}{l_i} \left(\frac{t}{l_i}\right)^{k_i - 1} \exp\left(-\left(\frac{t}{l_i}\right)^{k_i}\right) \tag{5}$$

$$(\boldsymbol{k}, \boldsymbol{l}, \boldsymbol{w}) = MLP_\theta(h_t) \tag{6}$$

Mixture of Fréchets The Fréchet distribution is also known as the Inverse Weibull and has similar properties to the Weibull in that it is defined on the positive reals, has shape (α) and scale (s) parameters and has a favorable form for the pdf and cdf which make it amenable for likelihood-based optimization. The primary distinction is that the Fréchet has heavy tails which can make it more stable for optimization purposes and more robust to outliers in the data. We define this mixture in a similar way:

$$p(t; \boldsymbol{s}, \boldsymbol{\alpha}, \boldsymbol{w}) = \sum_{i=1}^{J} w_i \frac{\alpha_i}{s_i} \left(\frac{t}{s_i}\right)^{-1-\alpha_i} \exp\left(-\left(\frac{t}{s_i}\right)^{-\alpha_i}\right) \tag{7}$$

$$(\boldsymbol{\alpha}, \boldsymbol{s}, \boldsymbol{w}) = MLP_\theta(h_t) \tag{8}$$

4.5. Encoding History with Multi-Channel LSTM

In order to encode history, we use a multi-channel LSTM architecture which is shown in Figure 2. Each event has its own dedicated LSTM which captures its irregular dynamics. The inputs to each LSTM are the time differences since the last observation of the event

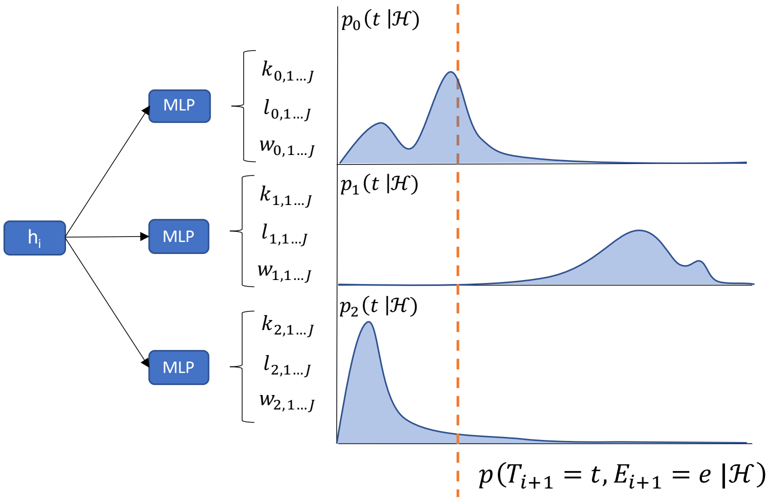

Figure 1: **Summary of the POPCORN Model**: The hidden encoding of history is mapped via MLPs to Weibull mixture parameters (mixture density network) and this can result in various conditional multi-modal distributions which are then used to compute the objective.

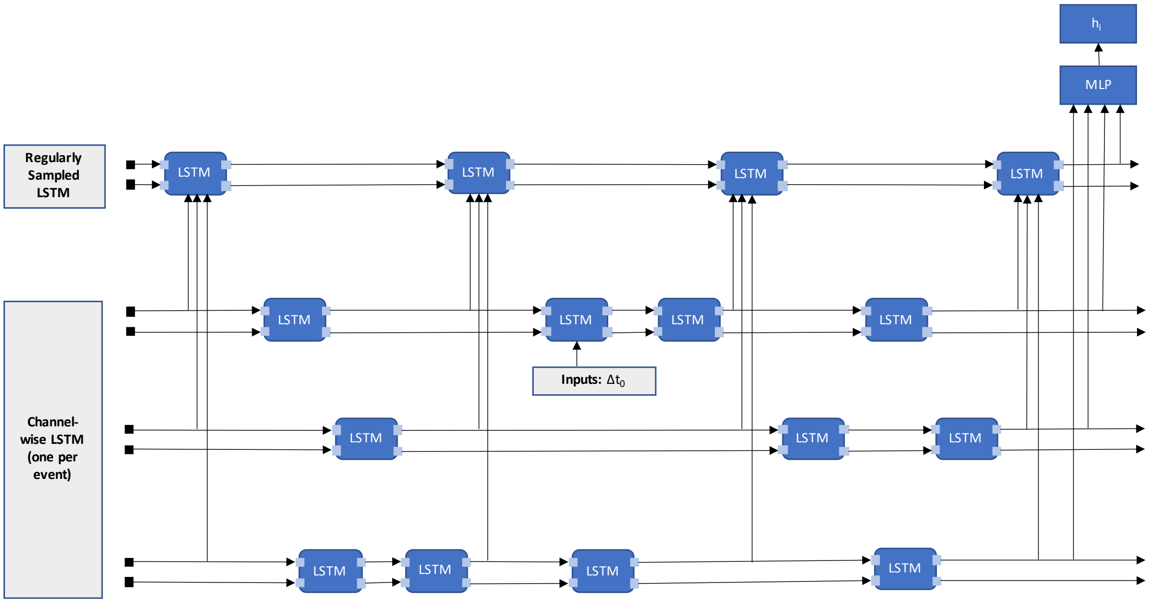

Figure 2: **Multi-Channel LSTM for encoding history**. Each event has its own distinct LSTM which keeps track of the progression of interevent times for that particular event. The regularly-sampled LSTM periodically collects the hidden states of all other channels to model dependencies across the channels overtime. At any given time, we may then collect all most recent hidden states, concatenate them and encode them as a single vector representing history.

$\Delta t = t_{e,i} - t_{e,i-1}$ where $t_{j,i}$ represents the absolute time of the ith observation of the jth event. Additionally, we have an LSTM channel dedicated to modeling dependencies across the rest of the channels over time. This LSTM takes as input the concatenated hidden states from each of the event-specific channels at a regular interval which is pre-specified. At each subsequent timestep, we can then extract all the hidden states of each LSTM and the regularly-sampled LSTM at that specific time and concatenate them. We use them as input into an MLP to create a hidden encoding of the history.

The basic motivation behind using such an approach is to capture the nature of the irregular sampling for each event. Additionally, such an approach may mitigate the problem of vanishing gradients especially for events which are rarely observed.

4.6. Handling Multiple Events at a Given Time

In general, multiple events at a given time are assumed to never occur in the context of point processes. EHR data, however, contains events which have the same timestamp largely as a function of documentation practices. We handle this by adjusting our objective function to allow a subset of the events to occur at a given time. We use an indicator vector to represent which events are present and which are absent.

The multi-label objective is as follows:

$$\begin{aligned} P(T_i = t, E_i = \mathbb{1}_e) &= \left(\prod_{e=1}^{|\mathcal{E}|} P(E_i = e | T_i = t)^{\mathbb{1}_e} (1 - P(E_i = e | T_i = t))^{(1-\mathbb{1}_e)} \right) P(T_i = t) \\ &= \prod_{e=1}^{|\mathcal{E}|} P(E_i = e, T_i = t)^{\mathbb{1}_e} (P(T_i = t) - P(E_i = e, T_i = t))^{(1-\mathbb{1}_e)} \qquad (9) \end{aligned}$$

This is effectively converting what was a categorical cross entropy to a binary cross entropy. We note that Enguehard et al. (2020) construct a similar loss for this situation. However, in their loss they are not modeling a conditional probability as we are but rather a joint probability as in Equation 1. They, instead, construct the following likelihood:

$$P(T_i = t, E_i = \mathbb{1}_e) = \prod_{e=1}^{|\mathcal{E}|} P(E_i = e, T_i = t)^{\mathbb{1}_e} (1 - \min(P(E_i = e, T_i = t), 1))^{(1-\mathbb{1}_e)} \qquad (10)$$

This likelihood assumes that the joint density can be treated as discrete and is constrained to be between 0 and 1. Thus, it requires bounding the joint density to compute. Due to this discrepancy and the lack of bound on the first term, it is difficult to compare our models on the NLL metric.

The general framework of point processes does not allow for simultaneous events. This likelihood provides one simple approach towards doing so. We note that there is related research (Solo, 2007) on how to handle ties in a more principled fashion and plan on incorporating these approaches in future work.

Table 1: Dataset Description

Dataset	Events	Task Type	Avg. Length	Train	Val	Test
MIMIC-II	75	Multi-class	4	585	65	65
Stack Overflow	22	Multi-class	72	5307	1326	1326
Ear Infection	39	Multi-label	2	8179	1022	1023
Synthea Full	357	Multi-label	43	10524	585	585

5. Experiments

5.1. Datasets

We run our experiments on four datasets in total. These are exactly the same datasets and dataset splits that were used in the work most closely related to our's (Enguehard et al. (2020)). We made the decision to use both common benchmarks used in the point process community and synthetic EHR data to encourage transparency and reproducibility. This also allows us to compare reported metrics directly.

MIMIC-II This is a dataset that has been used for benchmarking point processes methods in numerous past works. It consists of a sequence of hospital visits where each event is a different disease diagnosis. The average length of each sequence is relatively small (4) making this less of a longitudinal dataset than the full Synthea dataset.

Stack Overflow This dataset represents two years of user awards on a question-answering website. Each event is a user receiving a badge (of 22 different types) and when they received this badge. Although this dataset is not health related, it is used in almost every other point process paper as a benchmark and as such we used it to test the generalizability of our model.

Synthea: Ear Infection This dataset is simulated based upon the Synthea (Walonoski et al., 2018) EHR simulator which leverages a Markov process with several states informed by the input of human experts and population summary statistics. There are several modules in this simulator– this dataset leverages the Ear Infection module which is a simplified version of the full simulator that contains patients who experience ear infections. It consists of encounter types, conditions and medications associated with ear infections and any comorbidities associated with age of onset. This dataset is meant to be a simplified version of the full EHR simulation which has clear dependencies between time and next event.

Synthea: Full Simulation The full Synthea simulation consists of much longer longitudinal sequences (on average 43) of encounters, conditions and medications administered. Some of the most frequent events in this dataset include ER admission, viral sinusitis, insulin administration, and prenatal visits (among the 357 different event types).

5.2. Metrics

F1 and AUROC In order to evaluate how well our model does on next-event prediction, we use a weighted F1 score in the multi-class case (where only a single event can be observed

Table 2: Hyperparameter Settings for Reported Models

Dataset	Batch Size	Distribution	No. Mix	Hidden Enc	Hidden LSTM
MIMIC-II	16	Weibull	2	16	8
Stack Overflow	32	Fréchet	4	16	8
Ear Infection	16	Fréchet	2	16	8
Synthea Full	16	Weibull	4	16	8

at a given time) and weighted AUROC in the multi-label case (where multiple events can be observed at a given time). It should be noted that this is next-event prediction conditioned on the next time (as has been conventionally reported in past work).

Negative Log Likelihood We additionally report Negative Log Likelihood (NLL) normalized by time for the multi-class datasets (as this metric is not directly comparable with baselines for multi-label cases, see Section 4.6). The NLL is a measure of how well the model is capturing both time and event.

5.3. Hyperparameters

We list the most important hyperparameter settings in Table 2 which include batch sizes, distributional specification, number of mixture components, hidden embedding size and hidden size inside the channel LSTMs. We use the Adam optimizer with a learning rate of 1e-3 for all our runs, running every model for 100 epochs with early stopping criteria based on validation NLL.

6. Results and Discussion

Overall Findings Performance on the metrics is shown in Table 3 and Table 4 aggregated across five different splits, with sample standard deviation values over the splits in parenthesis. The results show that our model is able to achieve strong performance across all the datasets, particularly on next-event prediction. We compare our models against 4 baselines which are reported in Enguehard et al. (2020): Conditional Poisson (CP), RMTPP (Du et al., 2016), a Log Normal Mixture model (Shchur et al., 2019) and the best performing NeuralTPP model (Enguehard et al., 2020) for each dataset.

For the multi-class problems, our model performs competitively on F1 and NLL/time, achieving a better F1 score on the MIMIC-II dataset. For the multi-label case, our model performs equally well on AUROC on the Synthea Ear Infection dataset and significantly better on the full Synthea dataset over all baselines. As mentioned before, it is not possible to directly compare our results on the NLL/time metric as the likelihood functions are not exactly the same.

Our model outperforms CP, RMTPP and the LogNormMix on next-event prediction for all tasks. All of these baselines consider time and event independently. This provides strong evidence that for EHR data, incorporating this dependence is important. Furthermore, our assumption of conditional independence of event time distributions does not constrain

Table 3: Results on MIMIC and Stack Overflow

	MIMIC-II		Stack Overflow	
Model	F1 Score	NLL/time	F1 Score	NLL/time
CP	.691 (.083)	6.78 (1.99)	.325 (.004)	.553 (.003)
RMTPP	**.709 (.076)**	**4.24 (2.66)**	.284 (.004)	.592 (.006)
LogNorm Mix	**.705 (.170)**	6.33 (.370)	.314 (.003)	.548 (.004)
Neural TPP	.648 (.098)	**4.61 (2.49)**	**.342 (.006)**	**.543 (.005)**
POPCORN (Ours)	**.772 (.046)**	**5.07 (1.17)**	.330 (.005)	**.542 (.003)**

Table 4: Results on Synthea Datasets

	Synthea (Ear Infection)	Synthea (Full)
Model	AUROC Score	AUROC Score
CP	.792 (.009)	.850 (.014)
RMTPP	.675 (.068)	.616 (.043)
LogNorm Mix	.767 (.007)	.770 (.010)
Neural TPP	**.857 (.005)**	.822 (.006)
POPCORN (Ours)	**.853 (.008)**	**.886 (.008)**

model performance on next-event prediction as compared to the Neural TPP approaches which consider dependent competing risks. This provides some empirical evidence that for a sufficiently flexible specification of conditional distributions, we can effectively model EHR data despite this assumption. More work is needed to understand the effect of such an assumption on NLL / time.

AUROC by Event Type In order to examine which events the model is predicting best on the Synthea (Full) dataset, we visualize AUROC by event type in Figure 3. We can see that at a higher level, the model is able to predict medications most easily while conditions arehjo more difficult. In particular, for conditions which are potentially less predictable such as a concussion or appendicitis, the model does not perform as well. Medications which are commonly administered or prescribed for specific diseases (e.g. insulin for diabetes or furosemide for heart disease) are easier for the model to predict.

Model Performance by Length of History In order to evaluate our performance over long sequences of longitudinal data, we investigated how AUROC varied as a function of the number of observations seen on the Synthea (Full) dataset. We see in Table 5 that without any history and for shorter sequences, prediction is much more difficult. The longer the sequence, the more dependencies are able to be learned overtime. After collecting enough data about a particular patient's history (between 10-20 observations), the model is better able to reason about what comorbidities and medications a patient likely has and is likely to have in the future. The assumption of conditional independence is also mitigated by

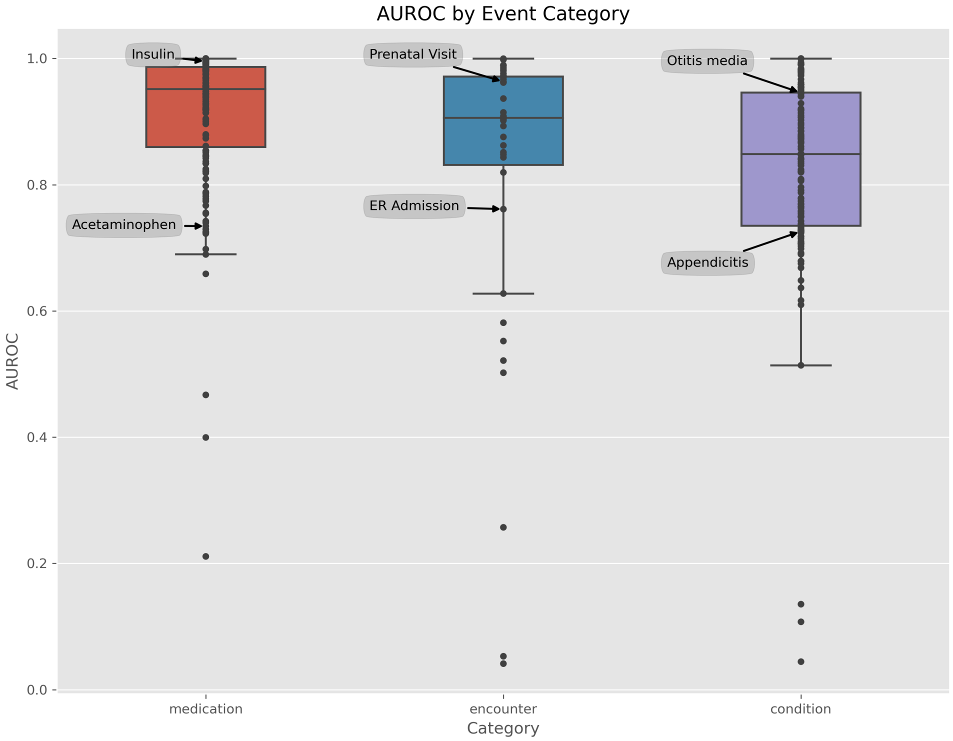

Figure 3: **AUROC by Event Type**: We observe that conditions are generally harder to predict than encounters and medications for the model.

Table 5: Performance by Sequence History Length on Synthea (Full)

Sequence Interval	0-1	0-5	0-10	0-20	0-30	0-40	0-50	0-60	0-70	0-80	0-90
AUROC	.601	.648	.723	.807	.835	.848	.857	.864	.869	.872	.876

the collection of more history which shows that for longer longitudinal sequences, such an assumption may be reasonable.

7. Conclusion

In this work, we presented a multivariate point process model for EHR data which has a number of advantages: (1) it specifies a dependence between event and time (2) allows for direct sampling and (3) specifies a closed-form likelihood, making optimization straightforward. We demonstrate that our approach matches or outperforms baseline approaches on the task of next-event prediction on all three clinical datasets. In particular, we outperform all baselines which do not take dependence between event and time into account for prediction. This dependence, while may be less important in certain datasets, is important to incorporate when modeling EHR data. Results also show that our model, which assumes conditional independence of event time distributions, performs similarly or better than NeuralTPP, as expected based on the theoretical results of Tsiatis (1975). Given the significant advantages (such as direct sampling and closed-form likelihood) that such an assumption enables, we believe that our approach should be strongly considered when such properties are particularly desirable. In future work, we aim to investigate different methods of handling ties which may reflect more closely the reality of the documentation process, evaluate our model on real longitudinal EHR data, and explore related applications such as encoding latent representations of history. Furthermore, we seek to evaluate our approach's ability to generate realistic samples of data and its performance on time-to-event with alternative metrics.

8. Limitations

Our study contains a number of limitations. We are unable to directly compare our approach to baselines (for multi-label scenarios) on the NLL metric. Additionally, we primarily leverage synthetic EHR data which is favorable from the perspective of reproducibility but represents a gap in evaluation which must be filled in future work by evaluating on real-world data. We also note that our approach to handling ties, while is simple and empirically performative, does not handle them in the most principled way. In future work, we look forward to addressing these limitations.

Acknowledgments

Many thanks to Joseph Enguehard et. al. for their work on Neural TPPs and for providing their code and data for community use in a very transparent manner. We aim to continue this tradition of transparency and provide our code as well. Thanks to the NLM Training Grant: T15LM007079-29 for funding this work.

References

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):1–12, 2018.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le 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*, pages 1555–1564, 2016.

Joseph Enguehard, Dan Busbridge, Adam Bozson, Claire Woodcock, and Nils Hammerla. Neural temporal point processes for modelling electronic health records. In *Machine Learning for Health*, pages 85–113. PMLR, 2020.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

Hongyuan Mei and Jason Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. *arXiv preprint arXiv:1612.09328*, 2016.

Maya Okawa, Tomoharu Iwata, Takeshi Kurashima, Yusuke Tanaka, Hiroyuki Toda, and Naonori Ueda. Deep mixture point processes: Spatio-temporal event prediction with rich contextual information. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 373–383, 2019.

Takahiro Omi, Naonori Ueda, and Kazuyuki Aihara. Fully neural network based model for general temporal point processes. *arXiv preprint arXiv:1905.09690*, 2019.

Jakob Gulddahl Rasmussen. Lecture notes: Temporal point processes and the conditional intensity function. *arXiv preprint arXiv:1806.00221*, 2018.

Oleksandr Shchur, Marin Biloš, and Stephan Günnemann. Intensity-free learning of temporal point processes. *arXiv preprint arXiv:1909.12127*, 2019.

Victor Solo. Likelihood functions for multivariate point processes with coincidences. In *2007 46th IEEE Conference on Decision and Control*, pages 4245–4250. IEEE, 2007.

Behzad Tabibian, Isabel Valera, Mehrdad Farajtabar, Le Song, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. Distilling information reliability and source trustworthiness from digital traces. In *Proceedings of the 26th International Conference on World Wide Web*, pages 847–855, 2017.

Matthew A Taddy, Athanasios Kottas, et al. Mixture modeling for marked poisson processes. *Bayesian Analysis*, 7(2):335–362, 2012.

Anastasios Tsiatis. A nonidentifiability aspect of the problem of competing risks. *Proceedings of the National Academy of Sciences*, 72(1):20–22, 1975.

Jason Walonoski, Mark Kramer, Joseph Nichols, Andre Quina, Chris Moesel, Dylan Hall, Carlton Duffett, Kudakwashe Dube, Thomas Gallagher, and Scott McLachlan. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. *Journal of the American Medical Informatics Association*, 25(3):230–238, 2018.

Appendix A. Baselines

All the baselines we compared against are described in detail in Enguehard et al. (2020). We describe them briefly as follows.

Conditional Poisson The conditional poisson model assumes that the event intensities are constant overtime (and thus assumes exponential event distributions specified by a parameter). This model also assumes that the next event and next time are conditionally independent.

$$\lambda^*(t) = MLP(h_t) \tag{11}$$

The MLP takes the historical encoding and transforms it to a constant which is then used to specify the closed-form likelihood.

RMTPP The RMTPP model (Du et al., 2016) uses the following specification:

$$\lambda^*(t) = exp\left(v^{t\top} h_j + w^t(t - t_j) + b^t\right) \tag{12}$$

$$P(y_{j+1} = k|h_j) = \frac{exp(V^y_{k,:}h_j + b^y_k)}{\sum_{k=1}^K exp(V^y_{k,:}h_j + b^y_k)} \tag{13}$$

where v^t (column vector), w^t (scalar), b^t (scalar) and V^y (matrix of size k by $|h_j|$) and b^y are all parameters of the model.

Additionally, h_j is the historical encoding which they obtain using an RNN which takes in tuples of the historical sequence.

Such a model has a more complicated intensity function than a conditional poisson but still requires the intensity to have an exponential formulation which results in a closed-form Gompertz likelihood. This model also models next events and times independently as shown above.

Log Normal Mixture The Log Normal Mixture model (Shchur et al., 2019) leverages a mixture distribution to directly model the event distribution as follows:

$$p(\tau|w, \mu, s) = \sum_{k=1}^{K} w_k \frac{1}{\tau s_k \sqrt{2\pi}} exp\left(-\frac{(log\tau - \mu_k)^2}{2s_k^2}\right) \tag{14}$$

where w are the mixture weights, μ are the mixture means and s are the standard deviations.

These mixture weights are parameterized by an embedding of past history as follows:

$$
\begin{aligned}
w_i &= softmax(V_w h_i + b_w) \\
s_i &= exp(V_s h_i + b_s) \\
\mu_i &= V_\mu h_i + b_\mu
\end{aligned}
$$

and $\{V_w, V_s, V_\mu, b_w, b_s, b_\mu\}$ are learnable parameters.

The next event is modeled independently:

$$
\pi_i = softmax(V_\pi^{(2)} tanh(V_\pi^{(1)} h_i + b_\pi^{(1)}) + b_\pi^{(2)}) \tag{15}
$$

where $\{V_\pi^{(1)}, V_\pi^{(2)}, b_\pi^{(1)}, b_\pi^{(2)}\}$ are parameters of the network and π_i is the categorical probabilities over the next events.

In order to encode history, they use the same architecture as RMTPP. The main advantage of this model over RMTPP is that they have a much more flexible distribution for the intensity (and thus for the event distributions).

Neural TPPs The Neural TPP models are a class of models which specify an encoder, decoder architecture. The encoder architecture encodes past history into a hidden vector and the decoder architecture specifies either (1) an analytical conditional intensity function for each event or (2) a cumulative conditional intensity function for each event. Within this framework, they have 2 encoder architectures and 4 decoder architectures which can be used interchangeably. For the encoders, they use either a standard GRU network or a Self-Attention (SA) network. For the decoder networks, they use either MLPs or attention networks to generate a conditional intensity or cumulative intensity. For further details, please refer to the appendix of Enguehard et al. (2020).

Appendix B. Synthea Dataset Details

Synthea Ear Infection As mentioned above, this dataset is simulated based upon the Synthea (Walonoski et al., 2018) EHR simulator which leverages a Markov process with several states informed by the input of human experts and population summary statistics. The ear infection module consists of encounter types, conditions and medications associated with ear infections and any comorbidities associated with age of onset. Table 6 shows all possible encounters/conditions/medications that are in this dataset along with their relative counts in a single fold of the training data.

Synthea Full Dataset The full Synthea simulation consists of much longer longitudinal sequences (on average 43) of encounters, conditions and medications administered. Table 7 includes the top 10 event names, types, codes and relative counts within each event category for a single fold of the training data.

Appendix C. Code

The code for the POPCORN model and data is provided at the following link: `https://github.com/sbhave77/POPCORN`.

Event Name	Event Category	Event Code	Count
Encounter for symptom	encounter	SNOMED-CT_185345009	10282
Otitis media	condition	SNOMED-CT_65363002	10282
Acetaminophen 160 MG Chewable Tablet	medication	RxNorm_313820	4384
Amoxicillin 250 MG Oral Capsule	medication	RxNorm_308182	2992
Aspirin 81 MG Oral Tablet	medication	RxNorm_243670	2972
Ibuprofen 100 MG Oral Tablet	medication	RxNorm_198405	2217
Penicillin G 375 MG/ML Injectable Solution	medication	RxNorm_105078	1713
Doxycycline Monohydrate 50 MG Oral Tablet	medication	RxNorm_1652673	912
Cefuroxime 250 MG Oral Tablet	medication	RxNorm_309097	871
General examination of patient (procedure)	encounter	SNOMED-CT_162673000	755
Ampicillin 100 MG/ML Injectable Solution	medication	RxNorm_789980	734
Cefaclor 250 MG Oral Capsule	medication	RxNorm_309045	645
Clopidogrel 75 MG Oral Tablet	medication	RxNorm_309362	590
Nitroglycerin 0.4 MG/ACTUAT Spray	medication	RxNorm_705129	424
Amoxicillin 500 MG Oral Tablet	medication	RxNorm_308192	406
Coronary Heart Disease	condition	SNOMED-CT_53741008	360
Simvastatin 20 MG Oral Tablet	medication	RxNorm_312961	348
Acetaminophen 325 MG Oral Tablet	medication	RxNorm_313782	347
Amlodipine 5 MG Oral Tablet	medication	RxNorm_197361	341
Stroke	condition	SNOMED-CT_230690007	307
Alteplase 100 MG Injection	medication	RxNorm_1804799	270
1 ML Epinephrine 1 MG/ML Injection	medication	RxNorm_1660014	265
Atropine Sulfate 1 MG/ML Injectable Solution	medication	RxNorm_1190795	265
Cardiac Arrest	condition	SNOMED-CT_410429000	265
History of cardiac arrest (situation)	condition	SNOMED-CT_429007001	257
3 ML Amiodarone hydrocholoride 50 MG/ML	medication	RxNorm_834357	251
Warfarin Sodium 5 MG Oral Tablet	medication	RxNorm_855332	211
Digoxin 0.125 MG Oral Tablet	medication	RxNorm_197604	211
Verapamil Hydrochloride 40 MG	medication	RxNorm_897718	210
Ibuprofen 200 MG Oral Tablet	medication	RxNorm_310965	202
Atrial Fibrillation	condition	SNOMED-CT_49436004	202
Well child visit (procedure)	encounter	SNOMED-CT_410620009	173
Naproxen sodium 220 MG Oral Tablet	medication	RxNorm_849574	160
Myocardial Infarction	condition	SNOMED-CT_22298006	144
History of myocardial infarction (situation)	condition	SNOMED-CT_399211009	134
Captopril 25 MG Oral Tablet	medication	RxNorm_833036	128
Atorvastatin 80 MG Oral Tablet	medication	RxNorm_259255	104
12 HR Cefaclor 500 MG Oral Tablet	medication	RxNorm_309043	55
Doxycycline Monohydrate 100 MG Oral Tablet	medication	RxNorm_1650142	48

Table 6: List of all possible events in Ear Infection dataset with event types and codes

Event Name	Event Category	Event Code	Count
Viral sinusitis (disorder)	condition	SNOMED-CT_444814009	32379
Acute viral pharyngitis (disorder)	condition	SNOMED-CT_195662009	19169
Normal pregnancy	condition	SNOMED-CT_72892002	16233
Acute bronchitis (disorder)	condition	SNOMED-CT_10509002	15901
Otitis media	condition	SNOMED-CT_65363002	8710
Streptococcal sore throat (disorder)	condition	SNOMED-CT_43878008	5616
Sprain of ankle	condition	SNOMED-CT_44465007	3641
Anemia (disorder)	condition	SNOMED-CT_271737000	2880
Body mass index 30+ - obesity (finding)	condition	SNOMED-CT_162864005	2750
Prediabetes	condition	SNOMED-CT_15777000	2062
Encounter for symptom	encounter	SNOMED-CT_185345009	89739
General examination of patient (procedure)	encounter	SNOMED-CT_162673000	72374
Encounter for check up (procedure)	encounter	SNOMED-CT_185349003	23610
Consultation for treatment	encounter	SNOMED-CT_698314001	23390
Emergency room admission (procedure)	encounter	SNOMED-CT_50849002	22673
Prenatal initial visit	encounter	SNOMED-CT_424441002	16233
Follow-up encounter	encounter	SNOMED-CT_390906007	13545
Encounter for problem	encounter	SNOMED-CT_185347001	11072
Encounter Inpatient	encounter	SNOMED-CT_183452005	7911
Well child visit (procedure)	encounter	SNOMED-CT_410620009	5988
Hydrochlorothiazide 25 MG Oral Tablet	medication	RxNorm_310798	27383
insulin human isophane 70 UNT/ML	medication	RxNorm_106892	20105
amLODIPine 5 MG	medication	RxNorm_999967	17760
Acetaminophen 325 MG Oral Tablet	medication	RxNorm_313782	17173
24 HR Metformin hydrochloride 500 MG	medication	RxNorm_860975	17170
Atenolol 50 MG Oral Tablet	medication	RxNorm_746030	16524
NDA020503 200 ACTUAT Albuterol 0.09 MG	medication	RxNorm_2123111	14255
120 ACTUAT Fluticasone propionate 0.044 MG	medication	RxNorm_895994	14255
Simvastatin 10 MG Oral Tablet	medication	RxNorm_314231	12214
Hydrochlorothiazide 12.5 MG	medication	RxNorm_429503	10509

Table 7: List of top 10 most frequent events by category in Synthea (Full) dataset with event types and codes