Temporal Logic Point Processes

Shen Yuan

- Abstract
- Temporal Logic
- Temporal Logic Point Processes
- Experiments

Abstract

This paper proposed a point process modeling framework based on temporal logic, which can predict when events would happen. It models the dynamics of the event by combinations of temporal logic formulae. In addition, this model can be considered as a general form of many well-known point processes like Hawkes process.

- Abstract
- Temporal Logic
- Temporal Logic Point Processes
- Experiments

First-order Logic

A predicate such as $Smokes(c)$ or $Friends(c, c')$, denoted as $x(\cdot)$, is a logic function defined over a set of entities $\mathcal{C} = \left\{c_1, c_2, \ldots, c_{|\mathcal{C}|}\right\}$, i.e.,

$$x(\cdot) : \mathcal{C} \times \mathcal{C} \ldots \times \mathcal{C} \mapsto \{0, 1\}\,. \tag{1}$$

A first-order logic rule is a logical connectives of predicates, such as

$$f_1 : \forall c, Smokes(c) \rightarrow Cancer(c);$$
$$f_2 : \forall c \forall c', Smokes(c) \wedge Friends(c, c') \rightarrow Smokes(c').$$

Temporal logic predicate

A temporal predicate is a logic function $x(\cdot,\cdot)$ defined over the set of entities $\mathcal{C} = \{c_1, c_2, ..., c_{|\mathcal{C}|}\}$ and time $t \in [0, \infty)$, i.e.,

$$x(c,t) : \mathcal{C} \times \mathcal{C} \ldots \times \mathcal{C} \times [0, \infty) \mapsto \{0, 1\}\,. \tag{2}$$

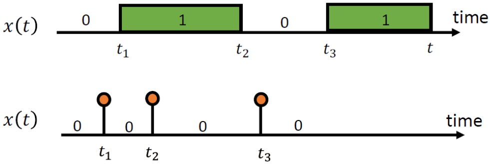


Figure: Illustration of grounded two-state (top) and point-based (bottom) temporal predicates.

Temporal relation

A temporal relation is a logic function defined as

$$r(\cdot) : (t_{A_1}, t_{A_2}, t_{B_1}, t_{B_2}) \mapsto \{0, 1\} \, . \tag{3}$$

where t_{A_1} and t_{B_1} are the starting times of intervals τ_A and τ_B, and t_{A_2} and t_{B_2} are the intervals' ending times.

A step function $g(s)$ and an indicator function $\kappa(s)$ can evaluate the temporal relation, which add hard temporal constraints,

$$g(s) = \begin{cases} 1, & s \geq 0 \\ 0, & s < 0 \end{cases} , \tag{4}$$

$$\kappa(s) = \begin{cases} 1, & s = 0 \\ 0, & o.w. \end{cases} , \tag{5}$$

Temporal relation

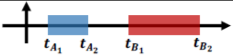
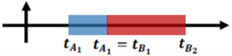
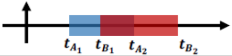
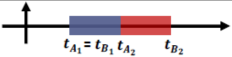
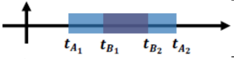
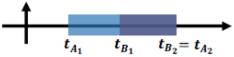
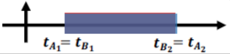
Temporal Relation	Logic Function $r(\cdot)$	Illustration
r_b: A before B	$g(t_{B_1} - t_{A_2})$	
r_m: A meets B	$\kappa(t_{A_2} - t_{B_1})$	
r_o: A overlaps B	$g(t_{B_1} - t_{A_1}) \cdot g(t_{B_1} - t_{A_2}) \cdot g(t_{B_2} - t_{A_2})$	
r_s: A starts B	$\kappa(t_{A_1} - t_{B_1}) \cdot g(t_{B_2} - t_{A_2})$	
r_c: A contains B	$g(t_{B_1} - t_{A_1}) \cdot g(t_{A_2} - t_{B_2})$	
r_f: A finished-by B	$g(t_{B_1} - t_{A_1}) \cdot \kappa(t_{A_2} - t_{B_2})$	
r_e: A equals B	$\kappa(t_{A_1} - t_{B_1}) \cdot \kappa(t_{A_2} - t_{B_2})$	

Figure: Function forms of 7 temporal relations.

Temporal logic formula

A temporal logic formula is a logical composition of temporal logic predicates and temporal relations, $f(\mathcal{X}_f, \mathcal{T}_f) \in \{0, 1\}$, where $\mathcal{X}_f = \{x_u(t)\}$ is a set of temporal predicates and $\mathcal{T}_f = \{\tau_u\}$is a set of time intervals.

$$f(\mathcal{X}_f, \mathcal{T}_f) := \left(\left(\bigvee_{x_u \in \mathcal{X}_f^+} x_u(t_u)\right) \bigvee \left(\bigvee_{x_v \in \mathcal{X}_f^-} \neg x_v(t_v)\right)\right) \\ \bigwedge \left(\bigwedge_{x_u, x_v \in \mathcal{X}_f} r_i(\tau_u, \tau_v)\right) \quad (6)$$

If the soft temporal relation constraints is used here, then the temporal logic formula $f(\mathcal{X}_f, \mathcal{T}_f) \in [0, 1]$.

- Abstract
- Temporal Logic
- Temporal Logic Point Processes
- Experiments

Intensity model for temporal predicate

A sequence of time intervals for each temporal predicate $\mathcal{H}_u(t)$ is defined as

$$\mathcal{H}_u(t) := \{x_u(0), x_u(t_1), x_u(t_2), \ldots, x_u(t_n), x_u(t)\} \tag{7}$$

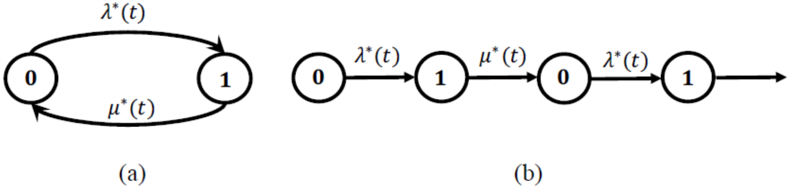


Figure: (a) Two-state transition diagram of a temporal predicate.(b) Unrolled conditional process.

Intensity design guided by temporal logic rules

$$\begin{aligned}\mathrm{FE} &= \delta_{f_1}(t|t_A \in \tau_A, t_B \in \tau_B) \\ &:= f_1(x_A, x_B, 1 - x_c, t_A \in \tau_A, t_B \in \tau_B, t_c = t) \\ &- f_1(x_A, x_B, x_c, t_A \in \tau_A, t_B \in \tau_B, t_c = t)\end{aligned} \tag{8}$$

where FE is short for formula effect, representing the transition intensity for x_c at any time t. The sign of FE can be 1, -1 or 0. The magnitude of the FE will quantify the strength of the influence. where

$$\begin{aligned}&f_1(x_A, x_B, x_c, t_A \in \tau_A, t_B \in \tau_B, t_c = t) \\ &= (\neg x_A(t_A) \vee \neg x_B(t_B) \vee x_C(t_C)) \wedge r_{be}(\tau_A, \tau_B) \wedge r_{be}(\tau_B, \tau_C)\end{aligned} \tag{9}$$

Intensity design guided by temporal logic rules

$$\lambda_C^*(t) = \exp\left\{\omega_{f_1} \cdot \sum_{\tau_A \in \mathcal{H}_A(t)} \sum_{\tau_B \in \mathcal{H}_B(t)} \delta_{f_1}(t | t_A \in \tau_A, t_B \in \tau_B)\right\} \quad (10)$$

where $\lambda_C^*(t)$ is the conditional transition intensity for x_C from state 0 to 1, and ω_{f_1} is a weight parameter that can be explained as the confidence level on the formula.

Intensity design guided by temporal logic rules

$$\lambda_C^*(t) = \exp\left\{\sum_{f\in\mathcal{F}_C} \omega_f \cdot \phi_f(t) + b(t)\right\} \tag{11}$$

$\mathcal{F}_C$ is a set of temporal logic formulae $\{f_1, \ldots, f_n\}$ for deducing $x_C(t)$.
$b(t)$ is a base temporal function.

Softened temporal constraints

$$g(s) = \min(1, \max(0, \beta s + \tfrac{1}{2})), \\ \textit{or}\ \ g(s) = \frac{1}{1 + \exp -\beta s}. \tag{12}$$

$$\kappa(s) = \max(0, \min(\tfrac{s}{\gamma^2} + \tfrac{1}{\gamma}, -\tfrac{s}{\gamma^2} + \tfrac{1}{\gamma})), \\ \textit{or}\ \ \kappa(s) = \frac{\exp(-|s|/\gamma)}{\gamma}. \tag{13}$$

Parameters β and $\gamma \geq 1$ can be either specified or treated as unknown parameters that will be learned from data.

Likelihood

$$\mathcal{L}(\{x_C(t)\}_{t\geq 0}) = \lambda_C^*(t_1) \exp\left(\int_0^{t_1} \lambda_C^*(s)\, ds\right) \cdot \mu_C^*(t_2) \exp\left(\int_{t_1}^{t_2} \mu_C^*(s)\, ds\right)$$
$$\ldots \exp\left(\int_{t_n}^{t} \mu_C^*(s)\, ds\right) \tag{14}$$

When the predicate x_C starts in state 0 and stays in state 1 up to time t.

- Abstract
- Temporal Logic
- Temporal Logic Point Processes
- Experiments

Recover Well-known Temporal Point Processes

Nonlinear Hawkes is a self-exciting point process with intensity function of the form

$$\lambda(t) = \exp(b + \alpha \sum_{t_i < t} \exp(-(t - t_i))), \tag{15}$$

where $b > 0$ and $\alpha > 0$, and it models the mechanism that previous events will boost the occurrence rate of new events. It can be expressed as

$$f_{\text{Hawkes}} : \mathit{Before}(t', t) \wedge (X_A(t') \to X_A(t)) \tag{16}$$

Recover Well-known Temporal Point Processes

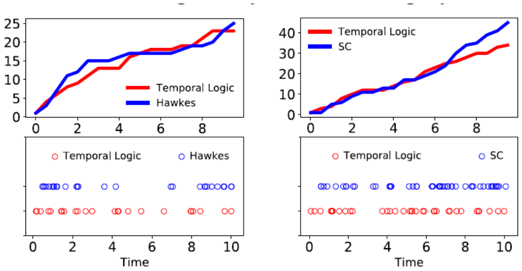
Self-Correcting point process is with intensity function of the form

$$\lambda(t) = \exp(bt - \sum_{t_i < t} \alpha), \tag{17}$$

where $b > 0$ and $\alpha > 0$, and it models the fact that previous events will inhibit the occurrence rate of new events. It can be expressed as

$$f_{\mathrm{self-correcting}} : \mathit{Before}(t', t) \wedge (X_A(t') \to \neg X_A(t)) \tag{18}$$

Recover Well-known Temporal Point Processes



(a): Hawkes. (b) Self-correcting.

Figure: Illustration of the generated events (in red) and the training events (in blue). Top: cumulative counts of the events. Bottom: realizations of the events that occur irregularly over time.

MIMIC-III: Predict Patient's Survival Rate

MIMIC-III is an electronic health record dataset of patients admitted to the intensive care unit. Authors had human experts define the logic rules according to the pathogenesis of sepsis.

Antibacterials	Use-Levofloxacin(t), Use-Ceftriaxone(t), Use-Meropenem(t), Use-Ceftazidime(t), Use-Tobramycin(t), Use-Metronidazole(t), Use-Vancomycin(t), Use-Azithromycin(t), Use-Ciprofloxacin(t), Use-Piperacillin(t)
Vasoactive Drugs	Use-Metoprolol(t), Use-Diltiazem(t), Use-Norepinephrine(t)
Diuretics	Use-Furosemide(t)
Symptoms	NormalBloodPressure(t), NormalHeartRate(t), NormalRespiratoryRate(t), NormalTemperature(t), NormalUrineOutput(t)
Suvival Condition	GoodSurvivalCondition(t)
Temporal Relation	Before(t, t'), Equal(t, t')

Figure: Defined Predicates for Sepsis Patients in MIMIC-III.

MIMIC-III: Predict Patient's Survival Rate

The first task is to predict the survival rate of each patient given her last recorded time in the dataset. All models were trained on training data containing 50, 500, and 4,000 patients' trajectories. The test data consists 100 patients.

Method	Train/Test: 50/100	500/100	4000/100
LSTM	0.405	0.420	0.436
RNN	0.439	0.442	0.424
LR	0.506	0.507	0.518
BN	0.530	0.570	0.540
Temp Logic	**0.584**	**0.647**	**0.682**

Figure: Predication Results of Survival Rate.

MIMIC-III: Predict Patient's Survival Rate

The second task is to predict the time to survival to validate continuous-time reasoning.

Method	Train/Test: 50/100	500/100	4000/100
RMTPP	0.464	0.398	0.399
Temp Logic	**0.233**	**0.252**	**0.227**

Figure: Prediction Results of Time to Survival: Mean Absolute Errors (MAE) in Days.

MIMIC-III: Predict Patient's Survival Rate

The third task is to evaluate the knowledge transfer ability via splitting MIMIC dataset based on age information of patients.

Method	Benchmark	Transfer1	Transfer2
RNN	0.511	0.507	0.517
LSTM	0.516	0.502	0.527
LP	0.519	0.514	0.522
BN	0.501	0.530	0.531
TempLogic	**0.615**	**0.637**	**0.659**

Figure: Prediction Results of Time to Survival: Mean Absolute Errors (MAE) in Days.