Forward sampling for State Variable Graphical Event Models with Markovian and piecewise constant dependencies

Moustafa Said Hawchar MOUSTAFA.SAID-HAWCHAR@UNIV-NANTES.FR
Nantes Université, CNRS, LS2N UMR 6004, F-44000 Nantes, France

Philippe Leray PHILIPPE.LERAY@UNIV-NANTES.FR
Nantes Université, CNRS, LS2N UMR 6004, F-44000 Nantes, France

Yanis Ouerdani YANIS.OUERDANI@ETU.UNIV-NANTES.FR
Nantes Université, CNRS, LS2N UMR 6004, F-44000 Nantes, France

Pierre Jannin PIERRE.JANNIN@UNIV-RENNES.FR
LTSI, INSERM UMR 1099, University of Rennes 1, F-35042 Rennes, France

Editors: Gustau Camps-Valls, Manuele Leonelli and Gherardo Varando

Abstract

Graphical event models are dedicated to modelling continuous-time processes. Some of their extensions are used to represent event occurrences and state variable changes w.r.t time. But the existing models in the literature struggle to capture complex historical dependencies between variables, limiting their applications in real-world processes. To address this limitation, we define a new class of state variable graphical event models (SVGEMs) that deals with more complex piecewise constant historical dependencies.

Sampling probabilistic graphical models is essential for approximate inference; previous studies have addressed this subject for several continuous-time models. However, sampling for graphical event models with various types of variables has not been explicitly described in the literature. To fill this gap, we also propose a forward sampling algorithm that unifies previous sampling algorithms into one single framework. A validation is performed with an experiment on randomly generated models, where we generate data using our sampling algorithm and show that the parameters of the models estimated from these data and priors converge to the ones of the underlying distribution. This work is the first one explicitly to describe a sampling algorithm for SVGEMs and pave the way for further research on the subject, including alternative SVGEM inference techniques such as importance sampling or Gibbs sampling.

Keywords: State variable graphical event model, Forward sampling, Parameter learning

1. Introduction

Several probabilistic graphical models have been proposed in the literature to represent dependencies in dynamic systems, dealing with time in a continuous or discrete way.

Dynamic Bayesian networks (DBNs) (Dean and Kanazawa, 1989) describe the evolution of state variables with discrete time when Continuous-Time Bayesian Networks (CTBNs) (Nodelman et al., 2002a) represent time continuously.

Some other continuous models have been proposed to describe temporal dependencies between events, such as Piecewise Conditional Intensity Models (PCIMs) (Gunawardana et al., 2011) and Graphical Event Models (GEMs) (Gunawardana and Meek, 2016). Several variants have been defined for GEMs: Timescale graphical event models (TGEMs)

© 2026 M. Said Hawchar, P. Leray, Y. Ouerdani & P. Jannin.

(Gunawardana and Meek, 2016), where the relation between two events depends on a set of timescales added to the edges in the form of half-open intervals: $(0, a], (a, b], ...$; or Proximal GEMs (PGEMs) (Bhattacharjya et al., 2018), with a proximal historical dependency based on one unique window $(0, t_h]$.

Few models have been suggested to address dependencies between events and state variables. In Event-driven continuous-time Bayesian networks (ECTBNs) (Bhattacharjya et al., 2020a), state variables can have state or event parents, when the event variables have only event parents. Recently, Bhattacharjya et al. (2020b) has proposed a more generic framework, State variable GEMs (SVGEMs), which represents dependencies between both types of variables. Even if this framework allows various kinds of historical dependencies for events or state variables, the authors focus on two specific historical dependencies (Markovian or proximal).

Exact probabilistic inference is possible for CTBNs but infeasible when the network size is large (Nodelman et al., 2002a). Several approximate methods have been introduced, based on sampling for CTBNs, GEMs or SVGEMs. Approximate inference using forward and importance sampling has been developed in CTBNs (Nodelman et al., 2002a), but also in models with event variables (Rajaram et al., 2005; Gunawardana et al., 2011). Gibbs sampling is proposed in PCIMs (Qin and Shelton, 2015).

One sampling method for ECTBNs is shortly mentioned but not described in (Bhattacharjya et al., 2020a). Another method is used for one specific class of SVGEMs, mixing approaches proposed for CTBN and PGEM in (Bhattacharjya et al., 2020b).

Our main contributions in this paper are:

1. a new class of SVGEMs with more complex piecewise constant historical dependencies,
2. a forward sampling algorithm for this class of SVGEMs, that combines sampling algorithms proposed for GEMs and CTBNs,
3. an experimental validation of this algorithm on randomly generated models, where we generate data using our sampling algorithm and show that the parameters of the models estimated from these data and defined priors converge to the ones of the underlying distribution.

2. State Variable Graphical Event Models

This section presents a formal definition of SVGEMs based on (Bhattacharjya et al., 2020b), and highlights various kinds of historical dependencies between state variables and events.

2.1. Notations

A temporal event sequence $\mathcal{D}_{\mathcal{E}} = \{(t_1, E_1),(t_2, E_2), \ldots, (t_{N_E}, E_{N_E})\}$ is a time-stamped sequence of labeled events (that belong to a finite set of event labels $\mathcal{E}$), with strictly increasing timestamps ($0 < t_i < t_{i+1} < t_{\text{end}}$), where N_E is the number of events and t_{end} the end of the sequence.

Discrete state variable variables are defined as $\mathcal{X} = \{X_j\}_{j=1}^{L}$, with $Val(X_j)$ the range of variable X_j. Let's define the multivariate discrete-valued time series $\mathcal{D}_{\mathcal{X}} = \cup_{X_j \in \mathcal{X}} \mathcal{D}_{X_j}$

where $\mathcal{D}_{X_j} = \{(t_k, x_k)\}_{k=0}^{k=N_X}$ is the sequence of transitions for state variable X_j observed until t_{end}.

Let $h(.)$ denote the historical occurrences for either event arrivals or state variable transitions. The history of events arrivals until time t for $\mathcal{E}$ is $h_{\mathcal{E}}(t) = \{(t_k, E_k) \in D_{\mathcal{E}} : t_k < t\}$, and the history of state variables transitions until t of the set $\mathcal{X}$ is $h_{\mathcal{X}}(t) = \{(t_k, x_k) \in D_{\mathcal{X}} : t_k < t\}$.

2.2. Model definition

Definition 1

Given a set $\mathcal{E}$ of event labels and a set $\mathcal{X}$ of state variables, an SVGEM is defined by a triplet $\mathcal{M} =< \mathcal{G}, P^0_{\mathcal{X}}, \Lambda >$. $\mathcal{G}$ is a graph defined over $\mathcal{E} \cup \mathcal{X}$ such that: (1) event variables $E \in \mathcal{E}$ have parents $U_E = \{U_E^{\mathcal{E}}, U_E^{\mathcal{X}}\}$ where $U_E^{\mathcal{E}}, U_E^{\mathcal{X}}$ are the event parents and the state parents, respectively, and (2) state variables $X \in \mathcal{X}$ have parents $U_X = \{U_X^{\mathcal{E}}, U_X^{\mathcal{X}}\}$, where $U_X^{\mathcal{E}}, U_X^{\mathcal{X}}$ are the event parents and the state parents respectively with $U_X^{\mathcal{X}} \subset \mathcal{X} \setminus \{X\}$. $P^0_{\mathcal{X}}$ is the initial distribution over state variables. $\Lambda = \{\Lambda_{\mathcal{E}}, \Lambda_{\mathcal{X}}\}$ are the conditional intensity parameters for the event labels and state variables.

Every event E occurs with a conditional intensity parameter $\lambda_{E|h_{U_E}(t)}$, where $h_{U_E}(t)$ is the history of all the occurrences in the parent set U_E until time t.

The inter-event time (waiting time) for the occurrence of E follows an exponential distribution with rate parameter $\lambda_{E|h_{U_E}(t)}$, similarly to marked point processes such as Poisson Networks (Rajaram et al., 2005) or PCIMs (Gunawardana et al., 2011). Λ_E is the set of the parameters of E for all possible parent's configurations and $\Lambda_{\mathcal{E}}$ is the set of all the Λ_E for $E \in \mathcal{E}$.

The dynamics of every discrete state variable X whose domain is $Val(X) = \{x_1, x_2, \cdots, x_n\}$ is described with $Q_{X|h_{U_X}(t)}$, the conditional intensity matrix (CIM) given the history of its parent set $h_{U_X}(t)$.

$$Q_{X|h_{U_X}(t)} = \begin{bmatrix} -q^X_{x_1|h_{U_X}(t)} \cdots & q^X_{x_1x_n|h_{U_X}(t)} \\ \vdots & \\ q^X_{x_nx_1|h_{U_X}(t)} \cdots & -q^X_{x_n|h_{U_X}(t)} \end{bmatrix} \tag{1}$$

with $q^X_{x_i|h_{U_X}(t)} = \sum_{j \neq i} q^X_{x_ix_j|h_{U_X}(t)}$.

$q^X_{x_i|h_{U_X}(t)}$ is the intensity of the state variable X leaving the state x_i (to another state) according to its parent history. $q^X_{x_ix_j|h_{U_X}(t)}$ is the intensity of transitioning from state x_i to x_j according to the parent U_X configuration. The waiting time for leaving the state x_i follows an exponential distribution with rate parameter $q^X_{x_i|h_{U_X}(t)}$. When leaving this state, the state variable will transition to its next state x_j with the probability $\theta^X_{x_ix_j|h_{U_X}(t)} = \frac{q^X_{x_ix_j|h_{U_X}(t)}}{q^X_{x_i|h_{U_X}(t)}}$.

Λ_X is the set of the parameters $q^X_{x_ix_j|h_{U_X}(t)}$ for all possible parent's configurations and $\Lambda_{\mathcal{X}}$ is the set of all the Λ_X for $X \in \mathcal{X}$.

2.3. Historical dependencies

Bhattacharjya et al. (2020b) proposes various kinds of historical dependence between variables $\mathcal{Z} = \mathcal{E} \cup \mathcal{X}$.

A historical dependence is **Markovian (M)** w.r.t set $\mathcal{Z}$, if the rate depends only on the most recent occurrence in history $h_{\mathcal{Z}}(t)$, denoted as $\widehat{h}_{\mathcal{Z}}(t)$.

A historical dependence is **piecewise constant (Pc)** w.r.t set $\mathcal{Z}$, if the rate at any time t depends on a specified state $\widetilde{h}_{\mathcal{Z}}(t)$ from a finite discrete state space mapped specifically from $h_{\mathcal{Z}}(t)$. Each possible history $h_{\mathcal{Z}}(t)$ is associated with one possible state $\widetilde{h}_{\mathcal{Z}}(t)$.

A historical dependence is **proximal (P)** w.r.t set $\mathcal{Z}$ if it takes into account the occurrence of the events or transitions of the state variables in $\mathcal{Z}$ in a time window $[t - t_h, t)$. This proximal dependence is a particular case of a piecewise constant one. For instance, with one single event parent E, $\widetilde{h}_{\mathcal{E}}(t)$ will be a boolean function describing that the event E has occurred or not in the time window $[t - t_h, t)$.

These historical dependencies can lead to the definition of several SVGEM classes **SVGEM_AB_CD**, where **A** and **B** are respectively the kind of historical dependencies chosen for state variable nodes w.r.t their state variable parents and their event label parents, and **C** and **D** define the same historical dependencies for the event label nodes.

For instance, two classes, **SVGEM_MP_MP** and **SVGEM_MP_PP**, are studied in (Bhattacharjya et al., 2020b). For **SVGEM_MP_MP**, the first **MP** states that the state variable nodes are Markovian w.r.t their state variable parents and proximal w.r.t their event label parents ; the second **MP** then expresses the same property for the event label nodes. For **SVGEM_MP_PP**, the historical dependence of the event label nodes is no longer Markovian w.r.t their state variable parents, but only proximal.

3. SVGEMs with Markovian and piecewise constant dependencies

3.1. Motivation

As represented in Section 2.3, the SVGEM framework has been mainly instantiated into two classes with proximal dependencies. Proximal dependence captures event occurrences based on recent history, since it is described for each edge with one simple time window $(0, t_h]$. However, this dependence is incapable of capturing complex relations where the occurrences of events vary depending on their parent's occurrences over multiple time windows.

In SPARS project[1], we are interested in describing surgical procedures (Lalys and Jannin, 2014) using SVGEMs. In a surgical sequence, each surgical event occurs after a certain amount of time following the previous ones. For example, a surgical event may happen less likely directly after its predecessor, and more likely after a certain amount of time. This specific kind of dependency can not be described with a simple proximal dependency.

For these reasons, we propose in this paper to develop one new class of SVGEMs, **SVGEM_MPc_MPc**, based on piecewise constant historical dependence instead of the simple proximal one.

1. https://project.inria.fr/spars/

3.2. Model definition

Definition 2 *SVGEM_MPc_MPc $\mathcal{M}$ is an SVGEM $< \mathcal{G}, P^0_{\mathcal{X}}, \Lambda >$ with parameters Λ defined as follows :*

$$\lambda_{E|h_{U_E}(t)} = \lambda_{E|\widetilde{h}_{U^{\mathcal{E}}_E}(t),\widehat{h}_{U^{\mathcal{X}}_E}(t)} \tag{2}$$

$$Q_{X|h_{U_X}(t)} = Q_{X|\widetilde{h}_{U^{\mathcal{E}}_X}(t),\widehat{h}_{U^{\mathcal{X}}_X}(t)} \tag{3}$$

where $\widetilde{h}_{U^{\mathcal{E}}_Z}(t)$ is a discrete projection defined from the history at time t of the event parents and $\widehat{h}_{U^{\mathcal{X}}_Z}(t)$ is the most recent configuration of the state variables parents observed at time t.

In this new class of SVGEMs, **SVGEM_MPc_MPc**, the variables (state variable or event label) are **Markovian** w.r.t their state variable parents and **piecewise constant** w.r.t their event label parent. These models can be seen as extensions (with state variables) of graphical event models such as PCIMs or TGEMs where the historical dependence between event variables is piecewise constant.

We will illustrate such models with an example of piecewise constant dependence defined in TGEMs (Gunawardana and Meek, 2016):

Definition 3 *An SVGEM_MT_MT $\mathcal{M}$ is an SVGEM_MPc_MPc where the piecewise constant historical dependence w.r.t event variables is defined by a set of timescales $\mathcal{T} = \{T_{E\to Z}\}$ corresponding to the edges going from an event label E to any variable Z. More specifically, a timescale is a set T of half-open intervals (time windows) $(a, b]$ that form a partition of some interval $(0, t_h]$. The piecewise constant historical dependence is defined by $\widetilde{h}_{U^{\mathcal{E}}_Z}(t) = h^c_{U^{\mathcal{E}}_Z}(t)$, the parent count vector, i.e., the (thresholded) number of occurrences for the event label parents of Z within every timescale interval of the corresponding edge.*

Usually, the threshold is fixed to one, which makes the count of the parent vector binary, absence or presence of the event labels parents of Z in every timescale interval.

Similarly, as discussed in Bhattacharjya et al. (2020b), these models can be expressed as GEMs where the state variable transitions are be represented as event variables, with Markovian and piecewise constant dependencies.

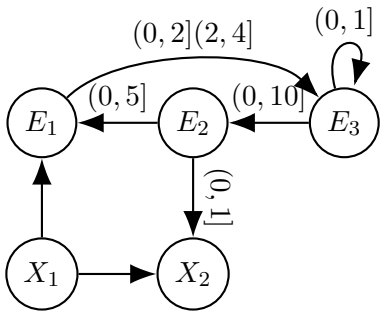


Figure 1: one example of SVGEM with Markovian and piecewise constant dependencies

Example 1 *Figure 1 describes an example of SVGEM_MT_MT with 2 state variables* $\mathcal{X} = \{X_1, X_2\}$ *and 3 event labels* $\mathcal{E} = \{E_1, E_2, E_3\}$.

Each state variable range is boolean : $Val(X_1) = \{x_1, \bar{x}_1\}$, $Val(X_2) = \{x_2, \bar{x}_2\}$.

Timescales are defined for every edge incoming from an event label, for instance $T_{E_2 \to X_2} = (0, 1]$, $T_{E_1 \to E_3} = (0, 2](2, 4]$ *and* $T_{E_2 \to E_1} = (0, 5]$.

Here the rate parameter of event E_1 *depends on the possible occurrence of* E_2 *in the time window* $[t-5, t)$ $(h^c_{E_2}(t) \in \{0, 1\})$, *and the most recent value observed for* X_1, $\hat{h}_{X_1}(t) \in \{x_1, \bar{x}_1\}$. *So the set* Λ_{E_1} *contains 4 parameters:* $\lambda_{E_1|0,x_1}, \lambda_{E_1|0,\bar{x}_1}, \lambda_{E_1|1,x_1}, \lambda_{E_1|1,\bar{x}_1}$.

Similarly, state variable X_2 *has also dependencies w.r.t state variable* X_1, *and the event label* E_2 *but with a different timescale* $(0, 1]$. *So its conditional intensity matrices defined for* X_2 *will depend on the possible occurrence of* E_2 *in the time window* $[t-1, t)$ *and the most recent value observed for* X_1 *:* $Q_{X_2|0,x_1}, Q_{X_2|0,\bar{x}_1}, Q_{X_2|1,x_1}, Q_{X_2|1,\bar{x}_1}$.

Let's notice that for boolean variables, this CIM is only defined by two parameters $[\ q^X_{x_1|h_{U_X}(t)}\ q^X_{\bar{x}_1|h_{U_X}(t)}]$ *and that* $\theta^X_{x_1\bar{x}_1|h_{U_X}(t)} = \theta^X_{\bar{x}_1 x_1|h_{U_X}(t)} = 1$ *(when you leave one state, you have no choice that reaching the other one).*

4. SVGEM sampling

4.1. Principle

Sampling algorithms have mainly been proposed for GEMs and CTBNs. As explained in (Gunawardana and Meek, 2016), the forward sampling algorithm proposed for Poisson networks (Rajaram et al., 2005) can be used for any piecewise constant temporal models such as PCIMs, TGEMs and PGEMs. This sampling algorithm relies on the decomposition of the joint distribution and progressively generates the value of each random variable in a topological order. As these models can have a complex graph with self-loops and cycles, this algorithm first identifies the strongly connected components (SCCs) of the graph and then generates data in topological order over the SCCs. In each SCC, occurrences of the interdependent events are generated with respect to the already generated events of the previous components.

The sampling algorithm proposed for CTBNs (Fan and Shelton, 2008) is quite different. All the state variables are sampled synchronously, but when one transition is selected for one of them, its next possible occurrence and the possible ones of its children are re-generated. Even if this algorithm seems quite different from the one proposed for GEM, we can show that both deal with their variables in the same order, as illustrated in Appendix A.

As **SVGEM_MPc_MPc** combines both event and state variables, the sampling algorithm we propose in the next section will generalize the one proposed for GEMs, generating iteratively each SCC (defined here with both events and state variables) and using the specificity of each kind of variable and their historical dependencies when generating the occurrences of the interdependent events and state variables inside one SCC. The use of SCCs is important because SVGEMs can contain self-loops. By decomposing the graph into SCCs, the overall structure is converted into a directed acyclic graph. This allows sampling to generate data in the topological order over the SCCs.

Algorithm 1: SVGEM-ForwardSampling

Input: $\mathcal{M} =< \mathcal{G}, P^0_{\mathcal{X}}, \Lambda >$, an SVGEM_MPc_MPc
Input: t_{end}, ending time of the sequence to sample
Output: $\mathcal{D}$, the sampled sequence

1 $\mathcal{D} \leftarrow \{(0, x_{01}), \cdots, (0, x_{0n})\}$
2 $SCCs \leftarrow extractSCCs_in_topological_order(\mathcal{G})$
3 **for** *each* $SCC \in SCCs$ **do**
4 $t \leftarrow 0$
5 **while** $t < t_{\text{end}}$ **do**
6 **for** *each* $Z \in SCC$ **do**
7 $(u_Z(t), \hat{t}_Z) \leftarrow get_parent_configuration(\mathcal{G}, Z)$
8 **if** $Z \in \mathcal{X}$ **then**
9 $\Delta t \leftarrow random.\text{Exp}(q^{z(t)}_{z_i|u_Z(t)})$
10 **end**
11 **else**
12 $\Delta t \leftarrow random.\text{Exp}(\lambda_{z(t)|u_Z(t)})$
13 **end**
14 $Time(Z) \leftarrow t + \Delta t$
15 **end**
16 $Z^* \leftarrow argmin_{Z \in SCC}(Time(Z))$
17 $\hat{t}^* \leftarrow min_{Z \in SCC}(\hat{t}_z)$
18 **if** $Time(Z^*) < min(t_{\text{end}}, \hat{t}^*)$ **then**
19 $t \leftarrow Time(Z^*)$
20 **if** $Z^* \in \mathcal{X}$ **then**
21 $z(t) \leftarrow random.\text{Mult}(\theta^{Z^*}_{z_i z_j|u^*_Z(t)})$
22 $\mathcal{D} \leftarrow \mathcal{D} \cup \{(t, z(t))\}$
23 **end**
24 **else**
25 $\mathcal{D} \leftarrow \mathcal{D} \cup \{(t, Z^*)\}$
26 **end**
27 **end**
28 **else**
29 $t \leftarrow min(t_{\text{end}}, \hat{t}^*)$
30 **end**
31 **end**
32 **end**
33 **return** $\mathcal{D}$

4.2. Sampling algorithm

Our forward sampling algorithm is described in Algorithm 1. The first step when sampling one sequence consists in drawing the initial state for all the state variables according to their initial joint distribution $P^0_{\mathcal{X}}$ (line 1). As explained in Section 4.1, the second step

is to extract the SCCs from our graph (line 2), and to sample our sequence in each SCC iteratively in a topological order.

The sequence sampling inside one SCC is then performed as follows :

1. We identify the configurations of the parents for each variable Z (line 7), $u_Z(t) = \{\widetilde{h}_{U_Z^{\mathcal{E}}}(t), \widehat{h}_{U_Z^{\mathcal{X}}}(t)\}$, and determine the next time $\hat{t}_z$ when this configuration changes, whether it was from an event occurrence or a state variable transition,

2. If $Z \in \mathcal{E}$, we generate the waiting time Δt_Z from the exponential distribution with rate parameter $\lambda_{z(t)|u_Z(t)}$. Otherwise, the waiting time Δt_Z is generated from the exponential distribution with rate parameter $q^{z(t)}_{z_i|u_Z(t)}$ (z_i is the state of the state variable Z at time t) (lines 8-14). The next state is generated from the multinomial distribution $\theta^Z_{z_i z_j|u_Z(t)} = \frac{q^{z(t)}_{z_i z_j|u_Z(t)}}{q^{z(t)}_{z_i|u_Z(t)}}$

3. Then we determine Z^*, the variable in this SCC that would occur first (line 16) and $\hat{t}^*$, the minimal time where the configurations of the parents remain unchanged (line 17).

4. If the sampled time for Z^* is valid ($Time(Z^*) < min(t_{\text{end}}, \hat{t}^*)$), then the sample is added in the sequence, and the current time t is updated to $Time(Z^*)$. Otherwise, the sample is rejected, and the current time t is updated to $min(t_{\text{end}}, \hat{t}^*)$).

4.3. Examples

Let us consider some simple examples with models depicted in Figure 2.

Example 2 ***Two independent nodes (E X):*** *Let us consider the simple case of a graph (Figure 2(a)) with an event label E and a boolean state variable X that are independent. So we have two independent SCCs (one for each node) that can be visited independently. When dealing with the SCC containing E, the waiting time Δt between two events follows an exponential distribution with a constant rate parameter λ_E. When dealing with the other SCC containing X, we have here a continuous finite-state homogeneous Markov process with $n = 2$ states, $X = x_0$ is chosen according to the initial distribution P^0_X; the transitions between states for the rest of the sequence are generated according to: (1) the waiting times*

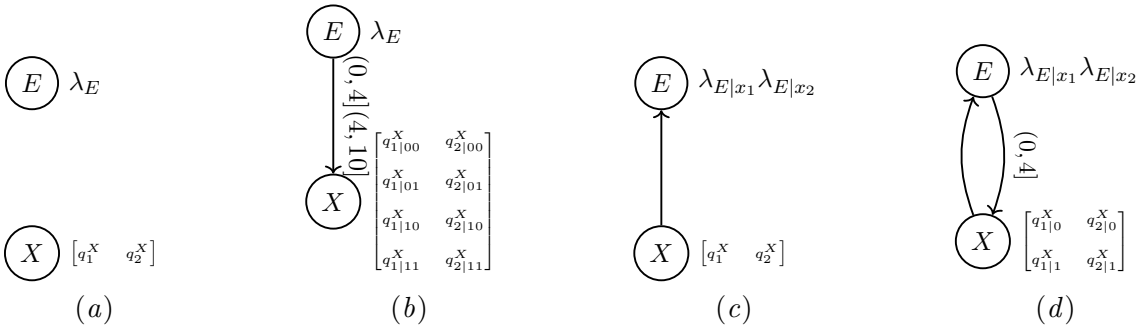


Figure 2: some examples of SVGEM_MT_MT models

Δt of staying in state x_i which follow the exponential distribution with a constant parameter $q^{x(t)}_{x_i|u_X}$, (2) and then X transitions to its other state.

Example 3 ***Two dependent nodes*** *($E \to X$): Let us consider the graph of Figure 2(b). There are yet two SCCs (one for each node) but they can not be visited independently. We must start sampling the full sequence of the SCC containing E (which is the same as in the previous example) then the one containing X.*

As before, $X = x_0$ is chosen according to the initial distribution P^0_X. But now, The CIM for X is piecewise constant according to the event E arrivals as described in Section 3. The rate parameters $[q^X_1 q^X_2]$ that generates the transitions between states are no more constant, but dependent on the possible presence of E in $[t-10, t-4)$ and $[t-4, t)$ (so 4 possible configurations $00, 01, 10$ and 11).

At t=0, no event has been observed so the first Δt is generated with the initial parameter rate $q^X_{x_0|00}$. But we know that this configuration is only valid until the first arrival of E (whose sequence has already been sampled in the previous SCC) or the end of the time window. If Δt is valid, one state transition is decided at $t = 0 + \Delta t$ to its other state. This process is repeated until t_{end}, by selecting the right rate parameter depending on the presence of E in $[t-10, t-4)$ and $[t-4, t)$.

Example 4 ***Two dependent nodes*** *($X \to E$): Figure 2(c) represents the graph where the dependence is reversed between E and X. There are also two SCCs (one for each node). We start by sampling X (as in the first example) then E whose dependence w.r.t X is Markovian. The rate parameter that generates the waiting time between two E is no more constant, but dependent on the current value of X, and valid until the next transition of X. The sampling mechanism is then quite similar as in the previous example.*

Example 5 ***Two dependent nodes*** *($E \rightleftarrows X$): A more complex example is proposed in Figure 2(d), where there is a cyclic relation between E and X and one unique SCC containing both nodes. The sampling here cannot be done separately, except at $t = 0$ where the event E hasn't occurred yet and the state variable X takes an initial value x_0 according to P^0_X.*

When dealing with the sampling of interdependent nodes inside one SCC, sampling is performed on a first-come, first-served basis. The rate parameters of E and X are selected with respect of the current configuration of their parent in order to generate their corresponding waiting time Δt and possible arrival time $Time(Z) = t + \Delta t$. The one with the minimal arrival time is selected and added to the sequence if this arrival time is yet valid. And so on until t_{end}.

4.4. Related work

If our SVGEM_MPc_MPc doesn't have event labels, our model is a classical CTBN, and our sampling algorithm is an improved version of one proposed for such models in (Fan and Shelton, 2008) where we take explicitly into account the SCCs to decompose the sampling process.

If our SVGEM_MPc_MPc doesn't contain state variables, our algorithm is exactly the one proposed for GEMs (Rajaram et al., 2005).

In the case where there are no arcs from state variables to event labels, our SVGEM_MPc_MPc model is richer than the ECTBNs proposed by Bhattacharjya et al. (2020a) that deal with proximal historical dependencies, and our sampling algorithm can be used to replace the one mentioned but not detailed in their paper.

5. Experiments and Results

To validate the results of our sampling algorithm, we propose to learn the parameters from generated data based on several randomly generated SVGEM_MT_MT. After that, we compare the original and learned parameters of the models.

5.1. Dataset

We generated 30 random SVGEM_MT_MT graphs for use in the sampling pipeline. To control the structural diversity and complexity of these models, we imposed conditions on several generation parameters. Here, the complexity of a graph refers to the total number of parameters (Λ_E, Q_X) it contains. In particular, we fixed the minimum and maximum numbers of total event labels and state variables to 3 and 10, limited the maximum in-degree of each node to 4, and constrained the number of strongly connected components to 3.

The horizon t_h of each piecewise constant dependence is initially generated from a uniform distribution between 0.5 and 5. We then partition each timescale with a maximum of 4 intervals, by recursively and randomly splitting one interval or extending the last one. Finally, the rate parameters are generated from a uniform distribution between 0.1 and 5, and the transition distributions are generated from a uniform distribution in the same range.

We have generated 10 graphs of complexity 50, 10 graphs of complexity 80, and 10 graphs of complexity 150 while maintaining the main design constraints.

We have then used our sampling algorithm to generate log files from each of the 30 models containing $N = 20, 100, 150, 500, 1000$ and 5000 sequences with duration $t_{\text{end}} = 20$ (which is sufficient to explore the configuration space of the parent variables). The experiment was repeated $K = 5$ times for each value of N.

5.2. Evaluation and validation

Given a model and a sampled log, we learn the conditional intensity parameters Λ, q and θ using the expectation a posteriori (EAP) method, following the methodologies used in (Nodelman et al., 2002b; Bhattacharjya et al., 2023). This approach improves the robustness of the estimations against the rare appearance of parent configurations in the sampled logs.

The intensity parameters λ_E and q_x have a Gamma prior distribution $Gamma(n', d')$, where n' and d' are the effective counts and durations of the prior over the conditional intensity parameters. The prior for the multinomial distribution θ can be represented with the Dirichlet distribution $Dir(n'[xx'|u], \ldots, n'[xx''|u])$.

So the estimated parameters can be written as follows:

$$\hat{\lambda}_{E|u} = \frac{n[E|u] + n'[E|u]}{d[u] + d'[u]}, \quad \hat{q}^X_{x_i|u} = \frac{n[x_i|u] + n'[x_i|u]}{d[x_i|u] + d'[x_i|u]}, \quad \hat{\theta}^X_{x_i x_j|u} = \frac{n[x_i x_j|u] + n'[x_i x_j|u]}{n[x_i|u] + n'[x_i|u]} \tag{4}$$

where $n[.|u]$ is the number of events E or transitions from x_i to x_j when the parents are in their configuration u, $d[.]$ is the duration of one configuration, and the constraints $n[x_i|u] = \sum_{x_j} n[x_i x_j|u]$ and $n'[x_i|u] = \sum_{x_j} n'[x_i x_j|u]$.

Inspired by the priors defined in Bhattacharjya et al. (2023) for PGEMs, we have decided to choose one simple prior for each event variable, or state variable transition independently of its parents, as described in Equation 5 and to divide it for each parent configuration u or state variable configuration x_i, which gives us the conditional priors described in equations 7 and 8.

$$n'[E] = 1, \qquad d'[E] = d'[X] = t_{\text{end}}, \qquad n'[x_i x_j] = 1(\text{if } x_i \neq x_j), 0 \text{ otherwise} \tag{5}$$

$$n'[x_i] = \sum_{x_j} n'[x_i x_j] = |Val(X)| - 1, \qquad d'[x_i] = \frac{d'[X]}{|Val(X)|} \tag{6}$$

leading to :

$$n'[x_i x_j|u] = \frac{n'[x_i x_j]}{m} = \frac{1}{m}(\text{if } x_i \neq x_j), 0 \text{ otherwise} \qquad n'[x_i|u] = \frac{|Val(X)| - 1}{m}, \tag{7}$$

$$d'[x_i|u] = \frac{d'[x_i]}{m} = \frac{t_{\text{end}}}{m|Val(X)|}, \qquad d'[u] = \frac{d'[E]}{m} = \frac{t_{\text{end}}}{m} \tag{8}$$

where the total number of parents configuration for a node Z is $m = 2^{|U_Z^{\mathcal{E}}|} \times \prod_{X' \in U_Z^{\mathcal{X}}} |Val(X')|$.

In order to show that the parameters of the models estimated from these data and priors converge to the ones of the underlying distribution, we propose in Figures 3, 4, and 5, the evolution of the absolute error between the learnt and true parameters, with its mean value MAE_G and standard deviation σ_G over our 5 repetitions, across various graph complexity and sample sizes. A detailed description of how MAE_G and σ_G are calculated is presented in Appendix B.

The results show that the MAE_G decreases toward zero as the sample size N increases, while the standard deviation also diminishes and becomes marginal beyond $N = 1000$. This indicates that the estimation error becomes both smaller and more stable as more sequences are available. We observe the same trend across the majority of the graphs, except several graphs such as graph50_7, graph50_9, graph150_4, and graph150_10, where the MAE_G tends toward a constant specific value instead of converging toward zero.

Figure 6 shows a detailed evolution of MAE_G for each state variable and event label in the graph with higher N. We can see that all variable estimates converge to zero except the estimates of the state variable X_0, which converge slowly toward zero even with larger data size N due to parent configurations that are very rare to occur together and sometimes even missing in our datasets, leading to a parameter estimation that depends largely on the priors. The same behavior is observed for the other graphs in Appendix C.

Overall, these results show that with a sufficiently large number of sampled sequences, the estimated parameters converge toward the original ones for both simpler and more complex models, or to values defined by priors, with respect to the sparsity of certain parent configurations, confirming that the sampling algorithm generates data consistent with the original distributions.

6. Conclusion

We have presented in this paper a new class of SVGEMs, SVGEM_MPc_MPc with a more complex piecewise constant historical dependencies than the State Variable Graphical Event Models presented in (Bhattacharjya et al., 2020b).

We have then proposed a forward sampling which unifies previous forward sampling algorithms into one single framework and can be applied to various existing models (CTBNs, ECTBNs, SVGEMs) and generalized for other kinds of historical dependencies.

We intend to strengthen our experimentation by choosing more coherent priors during parameter learning. We are also interested in proposing more evolved sampling algorithms used in probabilistic inference, such as importance Sampling (Fan et al., 2010) or Gibbs sampling (Qin and Shelton, 2015) for SVGEMs, and in developing one structure learning algorithm for such models, based on the one proposed for TGEMs (Gunawardana et al., 2011). Finally, going back to the starting point of our study, we will apply our algorithms and models for learning surgical process models from surgical logs.

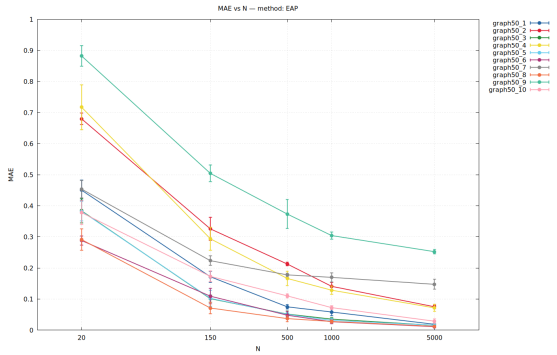


Figure 3: MAE_G and σ_G versus sample size N for graphs of complexity 50.

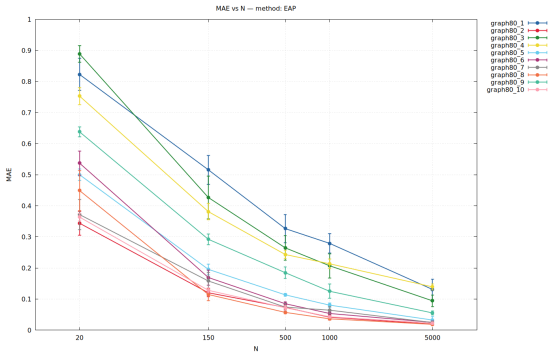


Figure 4: MAE_G and σ_G versus sample size N for graphs of complexity 80.

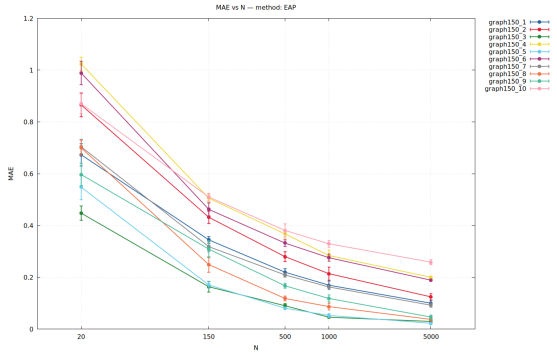


Figure 5: MAE_G and σ_G versus sample size N for graphs of complexity 150.

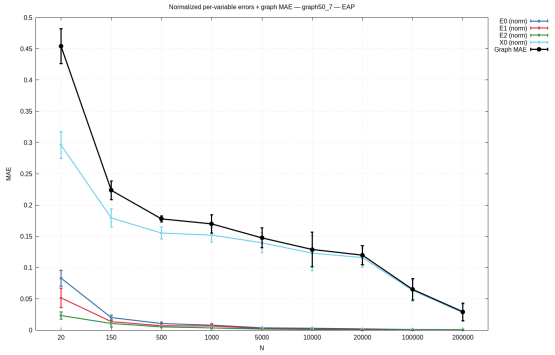


Figure 6: MAE_G and σ_G for the graph50_7 and its variables across various sample size N.

Acknowledgments

This work has received a French government support granted to the Labex Cominlabs excellence laboratory and managed by the National Research Agency in the "Investing for the Future" program under reference ANR-10-LABX-07-01.

References

Debarun Bhattacharjya, Dharmashankar Subramanian, and Tian Gao. Proximal graphical event models. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 8147–8156, 2018.

Debarun Bhattacharjya, Karthikeyan Shanmugam, Tian Gao, Nicholas Mattei, Kush Varshney, and Dharmashankar Subramanian. Event-driven continuous time Bayesian networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3259–3266, 2020a.

Debarun Bhattacharjya, Dharmashankar Subramanian, and Tian Gao. State variable effects in graphical event models. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4291–4297, 2020b.

Debarun Bhattacharjya, Tian Gao, Dharmashankar Subramanian, and Xiao Shou. Score-based learning of graphical event models with background knowledge augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12189–12197, 2023.

Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Computational intelligence*, 5(2):142–150, 1989.

Yu Fan and Christian R. Shelton. Sampling for approximate inference in continuous time Bayesian networks. In *Tenth International Symposium on Artificial Intelligence and Mathematics*, 2008.

Yu Fan, Jing Xu, and Christian R. Shelton. Importance sampling for continuous time Bayesian networks. *Journal of Machine Learning Research*, 11(72):2115–2140, 2010.

Asela Gunawardana and Christopher Meek. Universal models of multivariate temporal point processes. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 556–563, 2016.

Asela Gunawardana, Christopher Meek, and Puyang Xu. A model for temporal dependencies in event streams. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, page 1962–1970, 2011.

Florent Lalys and Pierre Jannin. Surgical process modelling: a review. *International Journal of Computer Assisted Radiology and Surgery*, 9(3):495–511, May 2014.

Uri Nodelman, Christian R. Shelton, and Daphne Koller. Continuous time Bayesian networks. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, page 378–387, 2002a.

Uri Nodelman, Christian R. Shelton, and Daphne Koller. Learning continuous time Bayesian networks. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, page 451–458, 2002b.

Zhen Qin and Christian R. Shelton. Auxiliary Gibbs sampling for inference in piecewise-constant conditional intensity models. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 722–731, 2015.

Shyamsundar Rajaram, Thore Graepel, and Ralf Herbrich. Poisson-networks: A model for structured point processes. *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, pages 277–284, 2005.

Appendix A. CTBN sampling algorithm

The sampling algorithm for CTBNs introduced in (Fan and Shelton, 2008) is described in Algorithm 2. To illustrate this algorithm, we consider a simple model with three state variables described in Figure 7.

First, the initial states for each state variable are chosen according to the initial distribution $P^0_{\mathcal{X}}$. Next, waiting times $\Delta t_1, \Delta t_2, \Delta t_3$ are sampled simultaneously for X_1, X_2 and X_3. Then, the state variable X^* with the minimum waiting time is selected. Depending on which time is minimum, one of the following scenarios occurs:

1. If X_1 has the minimum waiting time, it transitions to a new state. Its own time, as well as that of its child X_2, must be resampled (because of lines 17-22), while X_3's time remains unchanged.

2. X_2 has the minimum waiting time, it transitions to a new state. After that, its own time and that of its child X_3 are resampled, while X_1 is unchanged.

3. If X_3 has the minimum time, then the times of both X_2 and X_3 are resampled, while X_1 is unchanged.

From this, we observe that the sampling times of X_1 are resampled only if it undergoes a transition. So, X_1 can be sampled independently from the others, and, since it has no parents, the entire sequence that contains the transitions of X_1 until t_{end} can be sampled first. Otherwise, the waiting times for X_2 and X_3 are resampled whether they transition or one of their parents transitions to a new state. The sampling of a state for X_2 is influenced by the state of X_1 but not vice versa. So we can sample the sequence X_2 after the sequence of X_1. As X_2 and X_3 are connected, the sampling of their waiting times depends on each other states and therefore their sequences must be sampled synchronously.

In this organization of the sampling process, we recognize the SCCs of our model: X_1 forms one SCC, while X_2 and X_3 form another, and the SCCs are organized in a tree structure. So this CTBN sampling algorithm of CTBNs can be written with the logic of the sampling algorithm proposed by Rajaram et al. (2005) for GEMs.

Algorithm 2: CTBN Sampling (Fan and Shelton, 2008)

Input: $\mathcal{N} =< \mathcal{G}, \mathcal{Q}_{\mathcal{X}}, P^0_{\mathcal{X}} >$, a CTBN
Input: t_{end}, ending time of the sequence to sample
Output: $\mathcal{D}_{\mathcal{X}}$, the sampled sequence

$\mathcal{D}_{\mathcal{X}} \leftarrow \{(0, x_{01}), \cdots, (0, x_{0n})\}$
$t \leftarrow 0$
while *True* **do**
  for $X \in \mathcal{X}$ **do**
    if $Time(X) == \emptyset$ **then**
      $\Delta t \leftarrow random.\text{Exp}(q_{x_i|u_X})$
      $Time(X) \leftarrow t + \Delta t$
    end
  end
  $X^* \leftarrow argmin_{X \in \mathcal{X}}(Time(X))$
  if $Time(X) \geq t_{\text{end}}$ **then**
    return $\mathcal{D}_{\mathcal{X}}$
  end
  $t \leftarrow Time(X^*)$
  $x(t) \leftarrow random.\text{Mult}(\theta^*_{x_i x_j|u_{X^*}})$
  $\mathcal{D}_{\mathcal{X}} \leftarrow \mathcal{D}_{\mathcal{X}} \bigcup \{(t, x(t))\}$
  $Time(X^*) \leftarrow \emptyset$
  for $Y \in \mathcal{X}$ **do**
    if $X^* \in U_Y$ **then**
      $Time(Y) \leftarrow \emptyset$
    end
  end
end
return $\mathcal{D}_{\mathcal{X}}$

Appendix B. Calculation of the mean absolute error and standard deviation

Node level absolute error : For each variable Z, we compute an absolute error by comparing the learned parameters with the original parameters used to generate the data.

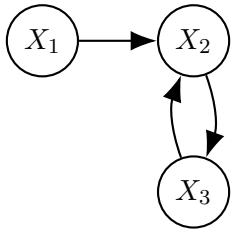


Figure 7: CTBN model $\mathcal{N}_1$

If $\alpha_{z,j}$ denotes the j-th original parameter of variable Z and $\hat{\alpha}^{(r)}_{Z,j}$ the corresponding estimate obtained at repeat $r \in \{1, \ldots, K\}$ for a given sample size N, then the absolute error of Z in repeat r is defined as

$$\mathrm{AE}^{(r)}_E \;=\; \frac{1}{m}\sum_{j=1}^{m}\left|\hat{\alpha}^{(r)}_{E,j} - \alpha_{E,j}\right|, \qquad \mathrm{AE}^{(r)}_X \;=\; \frac{1}{J}\sum_{j=1}^{J}\left|\hat{\alpha}^{(r)}_{X,j} - \alpha_{X,j}\right| \tag{9}$$

For an event label E, all estimated parameters λ are included, so we divide by the total number of parameters m.

For a state variable, we included the diagonal transition intensity rates q_X. As for the transition probabilities $\theta^X_{x_i x_j}$, since their sum on each row of the matrix is equal to one, only independent parameters were included. Therefore, one probability was omitted per row since it can be deduced from the others $\theta' = 1 - \sum \theta$. So the number of retained parameters per line is $|Val(X)| - 1$, and per matrix is $|Val(X)| \times (|Val(X)| - 1)$, and that gives us the total number for state variable $J = |Val(X)| \times (|Val(X)| - 1) \times m$

Graph-level absolute error : We then aggregate these node-level errors over all nodes of the graph. The absolute error of the whole graph for repeat r is defined as

$$\mathrm{AE}^{(r)}_G \;=\; \frac{1}{|\mathcal{Z}|}\sum_{Z\in\mathcal{Z}}\mathrm{AE}^{(r)}_Z. \tag{10}$$

where $|\mathcal{Z}|$ is the number of all the variables in the model.

This quantity measures the average parameter estimation error over the entire graph for one repetition of the sampling algorithm. For a fixed sample size N, we obtain K graph-level absolute errors $\mathrm{AE}^{(1)}_G, \mathrm{AE}^{(2)}_G, \ldots, \mathrm{AE}^{(K)}_G$. We then compute their mean and standard deviation:

$$MAE_G \;=\; \frac{1}{K}\sum_{r=1}^{K}\mathrm{AE}^{(r)}_G \tag{11}$$

and

$$\sigma_G \;=\; \sqrt{\frac{1}{K-1}\sum_{r=1}^{K}\left(\mathrm{AE}^{(r)}_G - \mathrm{MAE}_G\right)^2}. \tag{12}$$

Appendix C. Experimental results for several models with rare parent configurations

Figures 8, 9, 10, and 11 propose a detailed evolution of MAE_G for each state variable and event label in the corresponding graph. We can see that all parameter estimates converge to zero except the estimates of the event variable E_2 in Figure 8 and 9, variables E_1 and E_5 in Figure 10, and variable E_0 in Figure 11, which demands more data to converge toward zero ($N = 30000, 100000$).

A study of the corresponding variables shows us that some of their parent configurations are extremely rare, leading to a parameter estimation that depends largely on the priors.

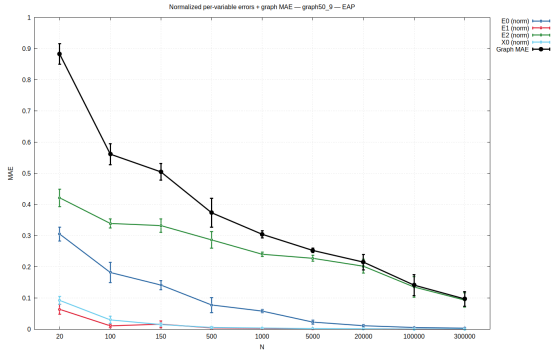


Figure 8: MAE_G and σ_G for the graph50_9 and its variables across various sample size N.

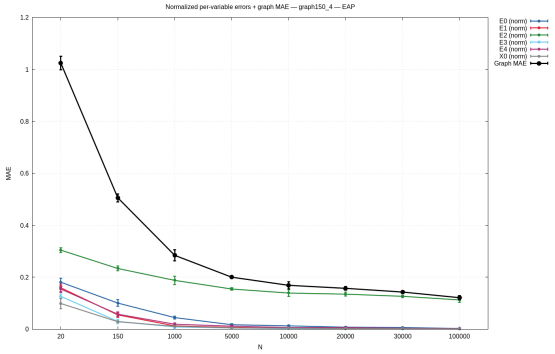


Figure 9: MAE_G and σ_G for the graph150_4 and its variables across various sample size N.

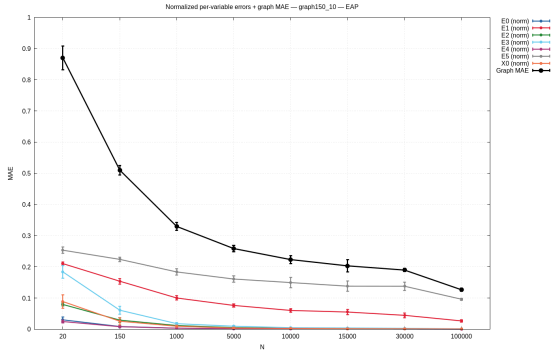


Figure 10: MAE_G and σ_G for the graph150_10 and its variables across various sample size N.

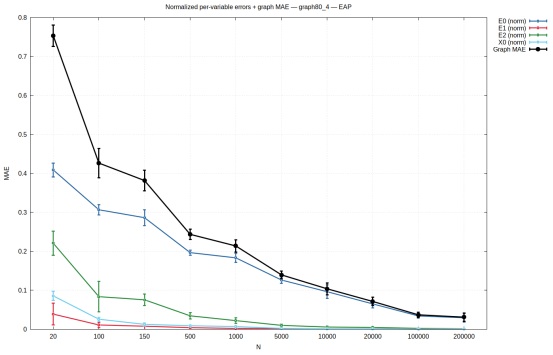


Figure 11: MAE_G and σ_G for the graph80_4 and its variables across various sample size N.