Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Marginalized Continuous Time Bayesian Networks for Network Reconstruction from Incomplete Observations

Lukas Studer
IBM Research Zurich
lukas.studer@alumni.ethz.ch

Loïc Paulevé
LRI UMR CNRS 8623
Université Paris-Saclay
loic.pauleve@lri.fr

Christoph Zechner
ETH Zurich
czechner@ethz.ch

Matthias Reumann
IBM Research Zurich
mre@zurich.ibm.com

María Rodríguez Martínez*
IBM Research Zurich
mrm@zurich.ibm.com

Heinz Koeppl*
Technische Universitaet Darmstadt
heinz.koeppl@bcs.tu-darmstadt.de

Abstract

Continuous Time Bayesian Networks (CTBNs) provide a powerful means to model complex network dynamics. However, their inference is computationally demanding — especially if one considers incomplete and noisy time-series data. The latter gives rise to a joint state- and parameter estimation problem, which can only be solved numerically. Yet, finding the exact parameterization of the CTBN has often only secondary importance in practical scenarios. We therefore focus on the structure learning problem and present a way to analytically marginalize the Markov chain underlying the CTBN model with respect its parameters. Since the resulting stochastic process is parameter-free, its inference reduces to an optimal filtering problem. We solve the latter using an efficient parallel implementation of a sequential Monte Carlo scheme. Our framework enables CTBN inference to be applied to incomplete noisy time-series data frequently found in molecular biology and other disciplines.

Keywords: sequential Monte Carlo; graph reconstruction; continuous time Bayesian network

1 Introduction

Influence network reconstruction from noisy, incomplete time series data is a challenging problem that has received a lot of attention, especially in biological sciences (Penfold and Wild 2011; Acerbi et al. 2014). Biological networks typically represent interactions between molecular components within the cell such as regulatory interactions among genes and transcription factors, or protein-protein interactions in signalling networks. Recent single-cell techniques allow to follow the activity of multiple molecular entities over cells and over time. The resulting ensemble of time-series can be modeled as noisy observations of a stochastic process, where each dimension of the state vector corresponds to a biological entity. Temporal changes in the molecular entities are reflected by transitions between different states of this process.

Continuous-Time Bayesian Networks (CTBNs) as proposed in (Nodelman, Shelton, and Koller 2002) model structured Markovian processes evolving in continuous time on a multivariate discrete state space. CTBNs are specified using a directed graph – not necessarily acyclic – over the system's variables and a set of rates for the state transitions of each variable that depends on the state of its parents. CTBNs are well suited as a hidden generative model for the biological processes mentioned above, because they can capture the stochastic effects that arise due to lowly abundant molecular species or other effects.

Prior applications of CTBNs to biological data (Acerbi et al. 2014) have assumed that the state of the system is known at all times through interpolating between data points in a pre-processing step. Assuming complete path observations is unrealistic for biological dataset but dramatically simplifies the inference problem: If additionally, the graph prior satisfies *structural modularity*, it can be shown that each node's incoming edges can be found independent of all other nodes' incoming edges (Nodelman, Shelton, and Koller 2003).

In absence of such simplifying assumptions, inferring a CTBN model from time-series data requires inferring the latent states of the system, which is a computationally challenging task as discussed in (Celikkaya and Shelton 2014; Fan, Xu, and Shelton 2010; Nodelman, Shelton, and Koller 2005). We extend these approaches by additionally considering measurements that are corrupted by noise, which is crucial for the biology domain. Several CTBN inference methods jointly infer both graph structure and rates, yet for biomedical applications knowledge about the presence of an (dysregulated) edge is often sufficient (Sonabend et al. 2014). To address this, we marginalize the CTBN's stochastic process description with respect to rates by extending the framework presented in (Zechner et al. 2014). This in turn gives rise to a substantial reduction in computational effort compared to standard techniques of joint inference and subsequent marginalization of the posterior.

* Co-last authors (and corresponding authors)

2 Background

We start out by defining CTBNs, introduce the applied measurement model and define the structure learning problem as a Bayesian model selection task.

2.1 Continuous Time Bayesian Networks

The CTBN model for discrete state, continuous time stochastic processes $\{X(t)\}_{t\geq 0}$ introduced in (Nodelman, Shelton, and Koller 2002) assumes that the discrete state space $\mathcal{X}$ is separable into a multivariate state space $\mathcal{X} = [\mathcal{X}_1, \ldots, \mathcal{X}_N]$. The state dynamics are then modeled by means of a directed dependence graph $\mathcal{G} = (V, E)$, where $V \equiv \{1, \ldots, N\}$ is the set of nodes (or variables) of the CTBN, and $E \subseteq V \times V$ is the set of edges. We denote by X_n the local state of the variable $n \in V$, with $X_n \in \mathcal{X}_n$. The transition probabilities of variable X_n depend only upon a subset of the other variables, referred to as the variable's parent set $\mathrm{dep}(n) \equiv \{m \mid (m, n) \in E\}$. The dynamics for the local state X_n are modeled as a conditional Markov process: Given the full state $U_n(t) = u$ of its parent set $\mathrm{dep}(n)$, X_n is a Markov process with transition intensities given by a conditional intensity matrix (CIM) $Q_n^u : \mathcal{X}_n \times \mathcal{X}_n \to \mathbb{R}$, that is

$$\Pr[X_n(t + \mathrm{d}t) = x' \mid X_n(t) = x, U_n(t) = u, Q_n^u] = \delta(x, x') + Q_n^u(x, x')\mathrm{d}t + o(\mathrm{d}t), \quad (1)$$

with $x, x' \in \mathcal{X}_n$ and δ the Kronecker delta. We refer to the set of all conditional intensity matrices by Q.

2.2 Measurement Model

We assume that several realizations of the stochastic process $X(t)$ are observed through noisy measurements which we assume to be drawn from a known distribution $p(y \mid X(t) = x)$ conditioned on the state $x \in \mathcal{X}$ at the time of measurement. A time-series data set $(Y, \boldsymbol{t})$ is a finite sequence of such measurements $Y = \{Y_1, \ldots, Y_L\}$ at time points $\boldsymbol{t} = \{t_1, \ldots, t_L\}$ with $0 < t_1 < \cdots < t_L$. That is, for all $1 \leq l \leq L$ it holds,

$$Y_l \sim p(y \mid X(t_l) = x), \quad (2)$$

with $x \in \mathcal{X}$ the realized state of the CTBN at time t_l. The measurement setup can be extended to more general incomplete data situations without affecting the proposed inference method.

2.3 Problem Statement and Model Selection

Measuring an ensemble of K time series data $\mathbf{Y} = \{(Y^1, \boldsymbol{t}), \ldots, (Y^K, \boldsymbol{t})\}$ of an unknown CTBN defined by $(\mathcal{G}', Q')$, we seek to infer the distribution over graphs $\mathcal{G}$ given $\mathbf{Y}$. Denote by $p(\mathbf{X}_t \mid Q, \mathcal{G})$ the path measure induced by the CIMs, with $\mathbf{X}_t$ a random path up to time t assuming values in the space of cadlag functions $D([0, t], \mathcal{X})$. Furthermore, denote by $\mathbf{x}_t^k \in D([0, t], \mathcal{X})$ the corresponding k-th sample path and $x^k(t_l)$ its evaluation at time t_l. With that, the problem statement reads

$$\begin{aligned} \text{Find} \quad & p(\mathcal{G} \mid \mathbf{Y}) \\ \text{Assuming} \quad & \mathbf{x}_\tau^k \sim p(\mathbf{X}_\tau \mid \mathcal{G}', Q') \\ & Y_l^k \sim p\big(y \bigm| x^k(t_l)\big) \end{aligned}$$

for all $1 \leq k \leq K$, $1 \leq l \leq L$ and $\tau \geq t_L$. As in earlier approaches to CTBN inference (Nodelman, Shelton, and Koller 2003) we seek MAP estimates of $p(\mathcal{G} \mid \mathbf{Y})$, by exploring the graph structure space, comparing graphs using their Bayesian score,

$$\mathrm{Score}(\mathcal{G}) = \ln p(\mathbf{Y} \mid \mathcal{G}) + \ln p(\mathcal{G}) \propto \ln p(\mathcal{G} \mid \mathbf{Y}),$$

where $p(\mathcal{G})$ is a prior on the graph structure. In the CTBN model, for a given node, there is one CIM for each combinatorial possible state of the node's parent set. Hence to regulate model complexity, the graph prior should penalize the number of incoming edges to each node in the model (Nodelman, Shelton, and Koller 2003).

3 Marginalized CTBN

We next introduce the marginalized process dynamics which we use to estimate the likelihood of time-series data with respect to a given CTBN graph structure using sequential Monte carlo methods.

3.1 Conditional Intensity Marginalization

In order to score a given graph configuration one traditionally computes the joint posterior over $\mathcal{G}$ and Q and subsequently marginalizes over Q. We present an alternative approach and marginalize the CTBN model directly. For the complete data scenario, it has been shown in (Nodelman, Shelton, and Koller 2003) that the path likelihood's Q-dependence can be marginalized out analytically, which allows to efficiently compute the marginal likelihood. In order to bypass the explicit inference of Q in case of incomplete measurements, we have to pursue a different strategy. In particular, we first construct a marginalized, parameter-free CTBN and subsequently evaluate the marginal likelihood $p(\mathbf{Y} \mid \mathcal{G})$ by solving the optimal filtering problem associated with this process and measurement model.

For illustration consider for now a single trajectory. We integrate $\Pr[X_n(t + \mathrm{d}t) = x', Q_n^u \mid X_n(t) = x, U_n(t) = u, \mathbf{X}_{t^-} = \mathbf{x}_{t^-}]$ with respect to Q_n^u, where we provisionally conditioned on the complete history of the process in the interval $[0, t)$. That yields

$$\begin{aligned} &\Pr[X_n(t + \mathrm{d}t) = x' \mid X_n(t) = x, U_n(t) = u, \mathbf{X}_{t^-} = \mathbf{x}_{t^-}] \\ &= \delta(x, x') + \mathsf{E}[Q_n^u(x, x') \mid \mathbf{X}_t = \mathbf{x}_t, \mathcal{G}]\mathrm{d}t + o(\mathrm{d}t) \quad (3) \end{aligned}$$

where we subsumed the conditioning at time t into the history $\mathbf{X}_t$ for the sake of conciseness and made explicit the dependency on the given graph $\mathcal{G}$. We emphasize that the resulting process $X_n(t)$ is parameter-free because the CIM in (1) is replaced by its estimate given the process history. Hence, the parameter uncertainty determined by the prior over the unknown CIMs is integrated into the process itself. The process accordingly becomes self-exciting. The result is analogous to the innovation theorem for counting processes (Aalen, Borgan, and Gjessing 2008). In order to be useful in practice, we need to find an explicit expression of this conditional expectation in terms of some path statistics of $\mathbf{x}_t$. To this end, note that by Bayes rule we have

$$p(Q \mid \mathbf{x}_t, \mathcal{G}) = \frac{p(\mathbf{x}_t \mid Q, \mathcal{G})p(Q \mid \mathcal{G})}{p(\mathbf{x}_t \mid \mathcal{G})}, \quad (4)$$

where $p(Q \mid \mathcal{G})$ denotes the prior over the entries of all CIMs given the underlying graph. The path likelihood $p(\mathbf{x}_t \mid Q, \mathcal{G})$ can be computed analytically by factorizing over the variables (Nodelman, Shelton, and Koller 2003). Accordingly, we introduce the summary statistics of the k-th path $\mathbf{x}_t^k$ for every node n and state u of the parent set $\mathrm{dep}(n)$: $T^u_{n,k}(x)$, $x \in \mathcal{X}_n$ the accumulated time spent in state x and $r^u_{n,k}(x, x')$ the count of state changes from state x to state x' within the interval $[0, t]$. The extension to K trajectories $\xi_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^K\}$ drawn from the same CTBN $(\mathcal{G}, Q)$ is achieved by noting that the trajectories are conditionally independent given Q. The factorization can then be collapsed over their joint summary statistics $T^u_n(x) = \sum_{k=1}^K T^u_{n,k}(x)$ and $r^u_n(x, x') = \sum_{k=1}^K r^u_{n,k}(x, x')$ as follows,

$$p(\xi_t \mid Q, \mathcal{G}) = \prod_{n=1}^{N} \prod_{u \in \mathcal{U}_n} \prod_{x \in \mathcal{X}_n} \prod_{x' \in \mathcal{X}_n \setminus x} \exp[Q^u_n(x, x') T^u_n(x)] Q^u_n(x, x')^{r^u_n(x, x')}.$$

Choices of prior distributions over Q that lead to tractable solutions of (4) are discussed in (Zechner, Deb, and Koeppl 2013). In this work we only consider independent univariate Gamma-type priors on the transition rates. We are free to choose its shape and rate parameters $\alpha^u_n(x, x')$ and $\beta^u_n(x, x')$ that are associated with a particular state transition and graph structure. This allows us to encode prior belief about the effect of edges in our model, e.g., activation or inhibition. A possible choice is discussed in section 3.4.

Given our choice of independent conjugate priors on the non-diagonal entries, it follows that the posterior (4) must also factorize over CIM entries $x, x' \neq x$ and each non-diagonal CIM entry is again Gamma distributed with parameters $\alpha^u_n(x, x') + r^u_n(x, x')$ and $\beta^u_n(x, x') + T^u_n(x)$. Consequently the expectation in (3) can be evaluated analytically,

$$\mathsf{E}[Q^u_n(x, x') \mid \xi_t] = \frac{\alpha^u_n(x, x') + r^u_n(x, x')}{\beta^u_n(x, x') + T^u_n(x)}. \tag{5}$$

With (5) we have an explicit form of the parameter-free marginal generator (3) and, conditionally on the path statistics, can draw samples of this process based on techniques for time-inhomogeneous Markov chains (Anderson 2007). The marginalized process $X(t)$ is no longer Markovian as it now depends on its history as well as the history of the other trajectories. However, by augmenting the state space by all summary statistics $\mathbf{T} = \{T^u_n(x) \mid n \in \{1, \ldots, N\}, u \in \mathcal{U}_n, x \in \mathcal{X}_n\}$ and $\mathbf{R} = \{r^u_n(x, x') \mid n \in \{1, \ldots, N\}, u \in \mathcal{U}_n, x \in \mathcal{X}, x' \in \mathcal{X} \setminus x\}$, the Markov property is recovered.

3.2 Marginal Likelihood Decomposition

The marginal likelihood required for scoring graphs as discussed in Section 2.3 is computed efficiently by exploiting its recursive structure. In particular, measurements can be taken into account one after another – either over time points and then trajectories or vice versa. In fact, one is free to choose the particular order of measurements as long as the resulting sequences are temporally causal. For instance, the third measurement of the ith trajectory cannot be processed before the second one and so forth.

Let $\mathcal{I}$ denote an ordered set of integer pairs $\{k, l\} \in \mathbb{N}^2$ observing this constraint. To simplify notation let $\mathbf{Y}^{k,l}_-$ denote the set of previously processed measurements excluding Y^k_l. Then it holds that

$$p(\mathbf{Y} \mid \mathcal{G}) = \prod_{(k,l) \in \mathcal{I}} p\Big(Y^k_l \;\Big|\; \mathcal{G}, \mathbf{Y}^{k,l}_-\Big). \tag{6}$$

For any single factor in (6), the law of total probability over the state X yields

$$p\Big(Y^k_l \;\Big|\; \mathcal{G}, \mathbf{Y}^{k,l}_-\Big) = \sum_{x \in \mathcal{X}} p\big(Y^k_l \mid x\big)\ \Pr\Big[X(t_l) = x \;\Big|\; \mathcal{G}, \mathbf{Y}^{k,l}_-\Big]. \tag{7}$$

The factors in the summands of (7) are the previously introduced measurement likelihood (2) and a prediction of the state distribution given earlier measurements. This recursive structure can be naturally exploited by employing Bayesian filtering techniques. Since this is analytically challenging for the marginal process model considered here, we resort to a sequential Monte Carlo (SMC) approach (Gordon, Salmond, and Smith 1993), in which the posterior distribution is computed by propagating a set of weighted particles.

Algorithm 1: Marginal Particle filter / Sequential Monte Carlo

Input: Measurement data $(Y, \boldsymbol{t})$, graph $\mathcal{G}$, initial set of M particles $\mathbf{p}^0$

Result: Estimate of log marginal likelihood $Z \approx \ln p(Y \mid \mathcal{G})$

for *measurement* $(y_l, t_l) \in (Y, \boldsymbol{t})$ **do**
  for *Particle* $p_m \in \mathbf{p}^{l-1}$ **do**
    `/* Prior prediction */`
    $p_m = \{x_m, \mathbf{T}_m, \mathbf{R}_m\} \leftarrow$ Propagate p_m through the marginal process model from t_{l-1} to t_l by sampling
    `/* Measurement likelihood */`
    $w_m \leftarrow p(y_l \mid X(t_l) = x_m)$
  end
  $Z_l \leftarrow \frac{1}{M} \sum_{m=1}^M w_m$
  $\mathbf{w} \leftarrow \mathbf{w} / Z_l$
  `/* Resample particles */`
  for *Particle* $p_m \in \mathbf{p}^{l+1}$ **do**
    $p_m \leftarrow$ Sample from $\mathbf{p}^l$ with probabilities $\mathbf{w}$
  end
end
$Z = \sum_{l=0}^L \ln Z_l$

3.3 Particle Filter

As indicated in the previous section, there are multiple ways of choosing the order in which measurements are incorporated during inference. For simplicity we consider the case

where whole trajectories are processed one after each other. The processing of a single trajectory $(Y, \boldsymbol{t})$ is illustrated in Algorithm 1. The algorithm requires to be initialized with a set of particles $\mathbf{p}_0 := \{p_m^0\} = \{\{X_m(0), \mathbf{T}_m, \mathbf{R}_m\} \mid m = 1, \ldots, M\}$, where $X_m(0)$ denotes the randomly drawn initial state of $X(t)$ and $\mathbf{T}_m$, $\mathbf{R}_m$ refer to the sufficient statistics stemming from the particle distributions associated with any preceding time-series and inference runs.

Note that the marginalized process dynamics as derived in the previous section is employed within SMC in the simulation step in Line 3 for the prior prediction.

3.4 Rate Prior

The marginalized prediction requires an assumption on the prior of the rate parameters given the graph structure, parent set state and transition, which we have fixed in Section 3.1 to independent Gamma distributions. In the reference implementation, all $\alpha_n^u(x, x')$ are set to 1, yielding exponential type priors, and $\beta_n^u(x, x')$ is set to favor activation type networks, that is, as the parent nodes state grows, the child node is also likely to transition into a higher state or remain in the highest state. Algorithm 2 gives the procedure in detail.

Algorithm 2: Activation-type network to β mapping

Input: Graph $\mathcal{G}$, prior hyper parameters $\hat{\beta} > 0, \tau > 0$ and $\epsilon \ll 1$, state pair x, x' where node n performs a transition, state of parent set u

Result: $\beta_n^u(x, x')$ rate parameter of rate prior distribution

```
/* Denote by u_i the i-th entry of u */
c_max ≡ sum of the parent set's states at their maximum
if |dep(n)| > 0 then
    c ← τ Σ_{i=1}^{|dep(n)|} u_i
    if x' > x then
        β̂ ← β̂ ln(1 + τc + ε)^{-1}
    else
        β̂ ← β̂ ln(1 + τ(c_max − c) + ε)^{-1}
    end
end
return β̂
```

4 Implementation

We briefly discuss key design choices for an efficient implementation targeting a high performance computing system with many compute nodes, each capable of executing several threads simultaneously.

4.1 Score Maximization

In order to maximize the score, similar to previous CTBN inference methods (Nodelman, Shelton, and Koller 2003), we use a greedy steepest ascent algorithm to explore the space of $2^{N(N-1)}$ graphs. In each step, all graphs in a Hamming distance 1 neighbourhood around the current best guess are scored, the next best guess being the graph with the best score. To deal with the significant increase in computational complexity owing to the incomplete, noisy data set, we employ distributed greedy search: Each evaluations of the scoring function is distributed amongst several processes. We opted to implement this step using `MPI`.

To deal with local minima and finite sample effects of the particle filter, we repeat the above algorithm several times, restarting from random initial configurations. This can also be trivially parallelized. Finally, we pool the best $W = 10$ graph configurations found by summing them weighted by their score, and then treat the resulting $[0, 1]^{N \times N}$ matrix as the marginal edge posterior. That is, having samples $\mathcal{G}^i \sim \Pr(\mathcal{G} \mid \mathbf{Y})$ with $\mathcal{G}^i = (V^i, E^i)$ we approximate the probability of an edge e, $\Pr(\mathbf{1}_E(e) \mid \mathbf{Y}) = \mathsf{E}[\mathbf{1}_E(e) \mid \mathbf{Y}]$ as the Monte-Carlo estimate

$$\sum_{\{\mathcal{G}=(V,E)\}} \mathbf{1}_E(e) \Pr(\mathcal{G} \mid \mathbf{Y}) \approx \frac{1}{W} \sum_{i=1}^{W} \mathbf{1}_{E^i}(e).$$

4.2 Parallelized Particle Filtering

Although the estimation of the marginal likelihood for the individual trajectories is coupled, at the cost of accuracy we may still opt to filter multiple trajectories simultaneously, sharing summary statistics whenever convenient, i.e., after finishing a batch of trajectories. We further approximate the sharing of summary statistics by computing their expected value after each trajectory for simplicity, which we found to work well in practice.

The above parallelization strategy requires us to keep full set of particles in memory for each compute process, which leads to memory bottlenecks for large graphs on multi core systems. Not only does the amount of required particles increase with the number of nodes, the same also holds for the number of summary statistics that need to be stored for each particle, that is, $|\mathbf{T}|+|\mathbf{R}| = \sum_{n=1}^{N} |U_n||\mathcal{X}_n|^2$, where $|U_n| = \prod_{i \in \mathrm{dep}(n)} |\mathcal{X}_i|$. Usually we will upper bound $|\mathrm{dep}(n)|$ via graph structure prior to i.e. 4, which for binary state spaces gives the upper bound of $\max_{\mathcal{G}} |\mathbf{T}| + |\mathbf{R}| = 64N$. To circumvent this memory bottleneck, we may opt to parallelize the particle filter's resampling (Murray, Lee, and Jacob 2014), forward simulation and measurement likelihood computation for which we only require a memory overhead of a single particle per thread. The particle filter lends itself more readily to shared memory type parallelizations with threads, using e.g. OpenMP or Pthreads.

5 Results and Discussion

The performance, validity and robustness of the proposed method is verified using synthetic data generated from CTBN models. Additional details concerning the synthetic data generation can be found in the supplementary materials[1].

We perform three series of experiments. A benchmark comparing the marginalized process filtering to alternative methods to score graphs, graph inference verification, and a robustness assessment.

[1] See http://www.bcs.tu-darmstadt.de/biocomm/hk/ctbn_aaai16_suppl.pdf

Experiment	Variation	AUROC	AUPR
Measurement noise	$\sigma = 0.2$	0.88	0.84
	$\sigma = 0.3$	0.86	0.80
	$\sigma = 0.4$	0.81	0.72
Inhibitory dynamics	$K = 10$	0.44	0.25
	$K = 20$	0.48	0.29
	$K = 40$	0.56	0.38
Boolean dynamics	1 Attractor	0.62	0.50
	> 1 Attractor	0.65	0.53

Table 1: Experimental results for robustness assessment with datasets generated from random CTBNs ($N = 5$), see supplementary materials[1] for details.

5.1 Particle Filter Benchmark

In order to score graphs, traditionally joint inference of both Q and $\mathcal{G}$ is performed, marginalizing first over the CTBN states, and then over Q. This is equivalent to state augmentation with the unknown CIM entries Q as static parameters, drawn from the Gamma distributed prior in the initial state. We refer to this approach as the *baseline* method. The baseline method suffers from sample impoverishment in the static parameters. This can be solved by resampling the static parameters Q after the measurement update, for each particle conditioned on its summary statistics, $p(Q \mid \mathbf{T}, \mathbf{R})$, as detailed in (Storvik 2002). Finally, one may marginalize out the Q as proposed in this paper.

To compare the three methods, numerical simulation was performed with single-threaded C++ implementations of all methods on an Intel i5-3450. The empirical standard error distribution of the estimated marginal likelihood is shown in Figure 1a), strongly suggesting that the standard error is the same for the resampling method and our method. However, when considering the compute time, our method outperforms the resampling method as shown in Figure 1b), hence justifying the increase in complexity due to the marginalization of the process.

5.2 Graph Inference Verification

The results discussed in this section and the next were obtained by considering the performance of the proposed algorithm over several graphs and conditional intensity matrices, except for a large graph exploration where only a single system is considered. For evaluation, we treat the graph recovery task as binary classification. We use the abbreviations ROC for Receiver Operator Characteristic, PR for Precision-Recall and the prefix AU to denote the area under either of the two.

The following two experiments verify that additional information improves the ability of our algorithm to recover the network structure in terms of accuracy.

Number of trajectories The exploration framework is given access to an increasing number of trajectories. From the results shown in Figure 2a), it is apparent that adding more data improves the ability of the proposed algorithm to recover the true graph as expected. Some of the spikes observed, i.e., the spike at 12 can be attributed to better parallel efficiency for those numbers of trajectories.

Additionally, we observe that in general, that for datasets that agree well with the prior assumption, the top ten graphs can be reached from many different original graphs as visualized in Figure 1c).

Gaussian noise sweep In this experiment, we vary the variance of the measurement noisy for the same underlying sets of trajectories. The exploration algorithm is given access to measurement noise standard deviation used to generate the respective data sets, i.e., the noise model parameters were not estimated. The results shown in Table 1 agree with intuition by favoring better quality data.

Large graph reconstruction We generated 40 time-series from a 11 node graph. We then ran the exploration algorithm on a single rack of Blue Gene/Q, consisting of 1024 compute nodes, each equipped with a PowerPC A2 CPU (16 compute cores, each 4-way multi-threaded). The results are shown in Figure 2b). On average, a local maxima is found every 2 minutes, we converge to a stable hypothesis after approximately 2 hours. This experiment confirms that the proposed method can recover networks of relevant size.

5.3 Graph Inference Robustness

In order to show that the proposed method is robust with respect to the assumptions encoded in the prior belief of the rates discussed in Section 3.4, we generate problem data from CTBNs with rates that do not reflect an activation type network.

Boolean networks The entries in the conditional intensity matrices may be chosen such that they emulate asynchronous Boolean networks see e.g. (Paulevé and Richard 2011). That is, for every configuration of the parent set of each node, a local attractor state is chosen. To generate a CIM from this set of local attractors, we fix the rate of all transitions to this attractor state to 0.9, while all other transition rates are set to 0.1. For nodes with no parents, a constant attractor is chosen. Such a procedure can easily lead to edges that do not influence the dynamics of their children, hence we developed the false negative edge penalty test to exclude such CIMs from the evaluation, as discussed in the supplementary materials.

For Boolean networks, it is possible to compute the full transition space and classify networks with respect to some dynamical features, notably their number of global attracting regions. Networks can exhibit a single attractor where all trajectories will eventually end up in; or several attractors, indicating either divergent trajectories, or disjoint reachable state spaces. In this paper we compare the performance of our exploration algorithm on systems that have a single global attractor region versus systems with multiple disjoint global attractor regions. The results shown in Table 1 imply that Boolean networks can be inferred, albeit with some difficulty.

Inhibitory dynamics Data is generated from an all inhibitory network, where incoming edges have an inhibitory

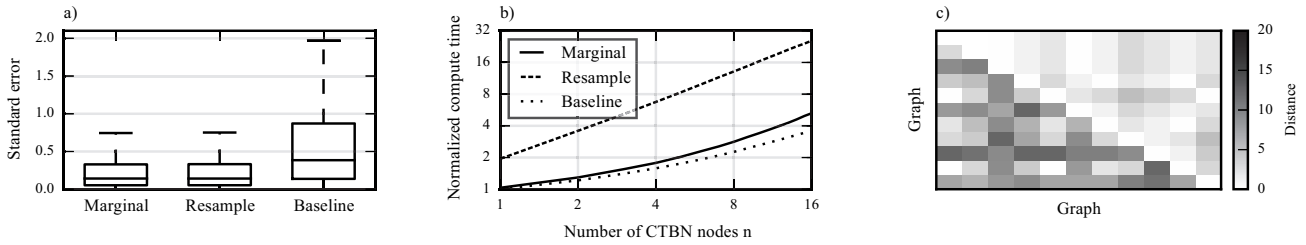

Figure 1: a) Standard error distribution of marginal likelihood, whiskers denote first and third quartile respectively, ground truth obtained via simulation with larger number of particles, b) Compute times over 1000 trajectories, averaged over 10 repeats, normalized to the time the baseline method took for the 1 node graph. The graph considered is a chain that is extended, i.e., $0 \to 1 \to 2 \to \ldots \to$ n, and c) Pairwise Hamming distances between the 10 best graphs obtained using greedy exploration (upper right) and their corresponding initial guesses (lower left), for a single exploration.

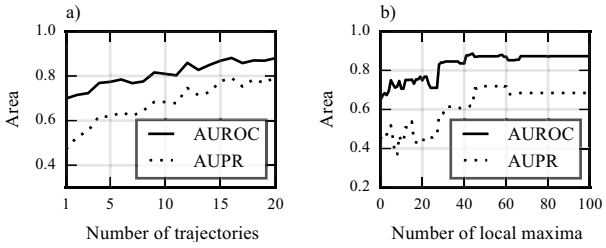

Figure 2: a) AUROC and AUPR for different number of trajectories, and b) Running AUROC and AUPR as local maxima are discovered for the large graph experiment.

effect. This is achieved by inverting the comparison in Algorithm 2, Line 4. Then similar to the first experiment, the exploration code is given access to varying numbers of trajectories. Given the choice of experimental parameters, the results are not satisfactory for purely inhibition networks, even when given access to many trajectories as shown in Table 1.

6 Conclusion

Motivated by applications in systems biology, this paper develops a method to recover a CTBN's graph structure from noisy, incomplete observations. Due to the secondary importance of the rate parameters, they are marginalized out analytically. For a given graph configuration, the resulting marginalized stochastic process represents the belief about all possible choices of CIMs under our prior assumption. We then close the loop by linking this prior assumption with the CTBN's dependence graph. This setup allows us to score graph configuration by solving the equivalent optimal filtering problem, which we do efficiently with sequential Monte Carlo. To deal with large graphs, we employ parallelized greedy search as well as parallelized variations of the particle filter. Compared to previous methods for fitting a CTBN to data, our method offers higher speed and accuracy by reducing the variance for a fixed compute budget, due to only inferring a specific part of the model. Finally the proposed method is verified and validated using synthetic data.

6.1 Future Work

For problems where nodes have many states (4 or more), the memory consumption can become the limiting factor in compute performance. We propose that the generality of the CTBN dynamics can be restricted by assuming that they behave similar to chemical reaction kinetics with linearly parametrized reaction rates, as the proposed marginalization still applies for the unknown rate parameter (Zechner et al. 2014). The benefit of such an aggregation is that when considering several transitions as a single type of reaction the number of sufficient statistics that have to be kept in memory is reduced as each reaction has only a single reaction count associated with it. In practice, such an aggregation can be achieved by changing the semantics of edges, i.e., if the type of reaction is assumed to be a birth-death process with birth rate proportional to the expression levels of the parents and a death rate proportional to its own state similar to what is proposed in (Äijö and Lähdesmäki 2009).

Acknowledgements

We would like to thank the IBM Data Centric Facilities and CCNI at Rensselaer Polytechnic Institute for allowing us to use their Blue Gene/Q for numerical experiments. CZ was supported by a grant from the Swiss SystemsX.ch initiative, evaluated by the Swiss National Science Foundation.

References

Aalen, O.; Borgan, O.; and Gjessing, H. 2008. *Survival and event history analysis: a process point of view*. Springer New York.

Acerbi, E.; Zelante, T.; Narang, V.; and Stella, F. 2014. Gene network inference using continuous time Bayesian networks: a comparative study and application to th17 cell differentiation. *BMC bioinformatics* 15(1):387.

Äijö, T., and Lähdesmäki, H. 2009. Learning gene regulatory networks from gene expression measurements using non-parametric molecular kinetics. *Bioinformatics* 25:2937–2944.

Anderson, D. F. 2007. A modified next reaction method for simulating chemical systems with time dependent propensities and delays. *The Journal of Chemical Physics* 127(21).

Celikkaya, E. B., and Shelton, C. R. 2014. Deterministic anytime inference for stochastic continuous-time Markov processes. In *Proceedings of the Thirty-First International Conference on Machine Learning*.

Fan, Y.; Xu, J.; and Shelton, C. R. 2010. Importance sampling for continuous time Bayesian networks. *Journal of Machine Learning Research* 11(Aug):2115–2140.

Gordon, N.; Salmond, J.; and Smith, A. 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)* 140:107–113(6).

Murray, L. M.; Lee, A.; and Jacob, P. E. 2014. Parallel resampling in the particle filter. *arXiv preprint* arXiv:1301.4019.

Nodelman, U.; Shelton, C.; and Koller, D. 2002. Continuous time Bayesian networks. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, 378–387.

Nodelman, U.; Shelton, C.; and Koller, D. 2003. Learning continuous time Bayesian networks. In *Proc. Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, 451–458.

Nodelman, U.; Shelton, C.; and Koller, D. 2005. Expectation maximization and complex duration distributions for continuous time Bayesian networks. In *Proceedings of the Twenty-first Conference on Uncertainty in AI (UAI)*, 421–430.

Paulevé, L., and Richard, A. 2011. Static analysis of Boolean networks based on interaction graphs: a survey. *Electronic Notes in Theoretical Computer Science* 284:93 – 104. Proceedings of The Second International Workshop on Static Analysis and Systems Biology (SASB 2011).

Penfold, C. A., and Wild, D. L. 2011. How to infer gene networks from expression profiles, revisited. *Interface Focus* 1(6):857–870.

Sonabend, A. M.; Bansal, M.; Guarnieri, P.; Lei, L.; Amendolara, B.; Soderquist, C.; Leung, R.; Yun, J.; Kennedy, B.; Sisti, J.; et al. 2014. The transcriptional regulatory network of proneural glioma determines the genetic alterations selected during tumor progression. *Cancer research* 74(5):1440–1451.

Storvik, G. 2002. Particle filters for state-space models with the presence of unknown static parameters. *Signal Processing, IEEE Transactions on* 50(2):281–289.

Zechner, C.; Unger, M.; Pelet, S.; Peter, M.; and Koeppl, H. 2014. Scalable inference of heterogeneous reaction kinetics from pooled single-cell recordings. *Nature Methods* 11(2):197–202.

Zechner, C.; Deb, S.; and Koeppl, H. 2013. Marginal dynamics of stochastic biochemical networks in random environments. In *European Control Conference (ECC)*, 4269–4274.