TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Elektrotechnik und Informationstechnik
Bioinspired Communication Systems

Bayesian Inference of Information Transfer in Networked Multi-Agent Systems

Master-Thesis
Elektro- und Informationstechnik

Eingereicht von

Gizem Ekinci

am
21.07.2020

1. Gutachten: Prof. Dr. techn. Heinz Koeppl
2. Gutachten: Dominik Linzner
3. Gutachten: Anam Tahir

Erklärung zur Abschlussarbeit gemäß §22 Abs. 7 und §23 Abs. 7 APB TU Darmstadt

Hiermit versichere ich, Gizem Ekinci, die vorliegende Arbeit gemäß §22 Abs. 7 APB der TU Darmstadt ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Mir ist bekannt, dass im Falle eines Plagiats (§38 Abs.2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden. Bei der abgegebenen Arbeit stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung gemäß §23 Abs. 7 APB überein.

English translation for information purposes only:

Thesis statement pursuant to §22 paragraph 7 and §23 paragraph 7 of APB TU Darmstadt:
I herewith formally declare that I, Gizem Ekinci, have written the submitted thesis independently pursuant to §22 paragraph 7 of APB TU Darmstadt. I did not use any outside support except for the quoted literature and other sources mentioned in the paper. I clearly marked and separately listed all of the literature and all of the other sources which I employed when producing this academic work, either literally or in content. This thesis has not been handed in or published before in the same or similar form. I am aware, that in case of an attempt at deception based on plagiarism (§38 Abs. 2 APB), the thesis would be graded with 5,0 and counted as one failed examination attempt. The thesis may only be repeated once. In the submitted thesis the written copies and the electronic version for archiving are pursuant to § 23 paragraph 7 of APB identical in content.

Darmstadt, den 21.07.2020

(Gizem Ekinci)

Abstract

Multi-agent systems in nature, such as a population of cells, cooperate through information sharing. This information can be incomplete and noisy. Inspired by such systems, we consider communication between three individuals, which are evolving continuously in time. Two of them act independently, and emit messages containing information about their states, while the third one receives a translation of these messages. Based on these translated observations, the agent node forms its belief over the state of other nodes. We modelled this system combining continuous-time Bayesian network (CTBN) and partially observable Markov decision (POMDP) process frameworks. The nodes evolve continuously in time as components of a CTBN. Given that the true messages are unavailable to the agent, the interaction between the nodes is modelled as POMDP. The exact update of the belief state is computed by filtering, and these results are used as a baseline. The approximation of the belief state is obtained using marginalized particle filtering. This work aims to infer the language model which leads to the translations.

Contents

List of Figures

List of Acronyms

AUPR	Area under Precision-Recall curve
AUROC	Area under Reciever-Operator-Characteristic curve
CIM	Conditional intensity matrix
CTBN	Continuous-time Bayesian network
CTMP	Continuous-time Markov process
FN	False negative
FP	False positive
FPR	False positive rate
MDP	Markov decision process
MP	Markov process
P	Precision
POMDP	Partially observable Markov decision process
PR	Precision-Recall curve
R	Recall
ROC	Reciever-Operator-Characteristic curve
SE	State estimator
SMC	Sequential Monte Carlo
TN	True negative
TP	True positive
TPR	True positive rate

List of Symbols

Notation	**Description**
$\Pr(\cdot)$	Probability
$p(\cdot)$	Probability distribution
$\mathrm{Gam}(\alpha, \beta)$	Gamma distribution with shape parameter α and rate parameter β
$w^\intercal$	Transpose of matrix/vector w
$\times$	Cartesian product
$\mathbb{E}_{p(x)}[f(x)]$	Expectation of $f(x)$ over probability distribution $p(x)$
δ	Kronecker delta
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Graph $\mathcal{G}$ with a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$
$\mathrm{Par}_{\mathcal{G}}(n)$	Parent set of node n in graph $\mathcal{G}$
$X(t)$	Value of random variable X at time t
Q	Transition intensity matrix
χ	State space of random variable X
$X^{[0,T]}$	Discrete valued trajectory of random variable X in time interval $[0, T]$
$\Upsilon(x_i)$	The total amount of time spent in state x_i in a trajectory
$r(x_i, x_j)$	The total number of transitions from state x_i to x_j in a trajectory
$r(x_i)$	The total number of transitions leaving state x_i a trajectory
ψ	Observation model
y	Observation
$\mathcal{Y}$	Observation space
a	Action
A	Action space
b	Belief state
π	Policy
ξ_T^K	Dataset of K sample trajectories with length of T
$\mathbf{k}$	Set of particles for particle filter

1. Introduction

A cooperative multi-agent system consists of relatively simple individuals, or agents, which come together to solve tasks in a way that will benefit the population. In such systems, the well-being of the population strongly benefits from the cooperation between the agents. The individuals might have limited abilities, however, they give rise to complex behaviour of the population through cooperation. These complex behaviours might be crucial for the survival of the population, e.g. running away from predators, rationing of vital substances, etc. Such systems can be found in nature even at the cellular level of living organisms [22] among many other examples, such as swarms [29].

Cells exist in stochastic environments. To maintain life, they are required to process noisy and fluctuating information coming from the environment and respond accordingly. In addition to the extracellular environment, the internal dynamics of cells are also found to be stochastic, such as their gene expressions [25]. Perkins and Swain (2009) [22] argue that, as the extracellular environment is a stochastic process, the intracellular processing of these stochastic signals and the choice of an appropriate response can only be probabilistic. This process consists of three main steps. The first step is to infer the current or a future state from the noisy signals. Then the cell must choose appropriate actions considering advantages and disadvantages conditioned on its inference. The final step is to take this action in a way that will contribute to the survival of the cell population.

Many studies take probabilistic approaches to explain the behaviour of cellular networks quantitatively. Statistical inference is presented as a possible framework to explain the mechanisms that a cell may use to interpret the state of the environment from noisy signals. Libby *et al.* (2007) [15] use Bayesian inference approach to model the gene expression of a bacterium in an environment with high and low levels of sugar, and show that the model is consistent with the measurements. This approach is later extended for the situations where the signal fluctuates over time, e.g. non-steady-state sugar concentration. Andrews *et al.* (2006) [1] propose that the cell adopts to such an environment by updating its beliefs in real-time. They model this decision-making mechanism as a sequential application of Bayesian inference, where the posterior probability that is inferred in the current step is used as a prior probability in the next step.

Bosher and Swain (2014) [2] take a similar approach to explain cellular decision making, focusing on information theory. They argue that if the mutual information between the signal to be inferred and the output of a signal transduction mechanism is high, only then the cell could be able to perform a high-quality inference. For example, Dubuis *et al.* (2013) [6], in their study on embryonic development of fruit flies, show that the mutual information between gap gene expression and the position of the nucleus is the information needed for each cell to determine their position along the long axis of the embryo.

The cooperation of the agents is optimized through the exchange of information between the agents. In honeybee and ant colonies, individuals share information about potential nest sites with the colony through recruitment signals, which are accumulated to acquire an intelligent collective decision on the best nest site [8]. In bacteria, individuals communicate by quorum sensing where they sense and produce hormone-like molecules called autoinducers. This process enables bacteria to observe the environment for the changes in the population or the presence of other bacteria and respond accordingly [31].

The influence of communication on cooperative behaviour in stochastic environments is an area of ongoing research. Paarporn *et al.* (2018) [21] study the optimal behaviour in a two-player cooperative game in a stochastic environment with and without information sharing in the presence of noise in communication channels. They find that the majority logic strategy is optimal when the agents have intermediate reliable information from the environment and the information sharing between them is highly reliable.

1.1. Motivation

Inspired by the examples of multi-agent systems in nature, we are interested in getting insights about communication in a system where individuals rely on the exchange of information to achieve optimal behaviour. To this end, we consider a communication problem between individuals which evolve continuously in time. The messages sent in the system are translated by a language model. The translations are observed by an acting agent which then performs some task in coordination with others. We assume that the behaviour of the agent has been shaped by evolution (close) to optimality and that the optimal behaviour is known. The goal of this work is to infer the language model from demonstrations.

As a motivating example to this work, consider the state of a cell expressed in terms of its gene activation state. Based on this gene state, the cell may emit some message. Another cell, which we referred to as agent, receives a translation of the messages emitted by its neighbouring cells. Given some task and the corresponding optimal policy, the agent cell gauges its gene state in coordination with its neighbours.

We consider three individuals participating in the communication, two of which share information, and an agent which processes these messages. We model the individuals as components of a continuous-time Bayesian network (CTBN). The states of the components are emitted as messages. The agent does not have access to these messages directly but observes a translation of them, which poses a partially observable Markov decision process (POMDP). The agent taking an action corresponds to changing its dynamics. Our dataset consists of discrete-valued state trajectories of the components in CTBN, from which we infer the language model, that leads to the translated messages. As it is modelled within the POMDP framework, we refer to the language model as observation model in the remainder of the thesis.

The observation model can be interpreted in several ways which provide different scenarios to the problem that we consider. It can be interpreted as the communication protocol, where the individuals choose to share limited information with the population based on their states. The model can also be considered as a noisy communication channel or a summary/aggregation imposed on the messages by the environment. In the following, we focus on the latter.

1.2. Contributions

The main contributions of this thesis are as follows:

- Formulation of the information transfer with a continuous-time graph-based approach
- Implementation of the system employing CTBN and POMDP frameworks and synthetic data generation
- Approximation of the belief state through marginalized CTBN
- Inference of the observation model from continuous-time demonstrations
- Evaluation of the performance on the synthetic data

1.3. Structure of the Thesis

The remainder of the thesis is structured as follows:

Chapter 2 presents the theoretical background of this work. It reviews the details of continuous-time Bayesian networks and partially observable Markov decision processes and introduces the sampling algorithms used in this work.

Chapter 3 is dedicated to the details of the problem. It explains how the frameworks are utilised, and presents the algorithms used for data generation and inference.

Chapter 4 presents the experimental results of the simulation and the inference.

Chapter 5 discusses the results, highlights the conclusions and reviews the limitations. Moreover, it gives suggestions for future directions and extentions to this work.

2. Foundations

This chapter presents the theory applied in this thesis. First, the details of the communication problem are described briefly to put the theory into perspective, and then the mathematical background of the frameworks used to model this problem is introduced.

2.1. Problem Formulation

The communication model considered in this work is given in Figure 2.1. The parent nodes X_1 and X_2 emit messages which contain information about their states. These messages are translated by an observation model ψ. An agent node X_3 makes a decision based on this translated message y. The main objective is to infer the observation model, given a set of trajectories of the nodes.

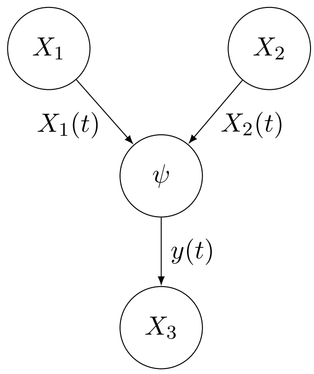

Figure 2.1.: The communication model.

The transition models of the nodes and the dependencies between them are modelled as a continuous-time Bayesian network (CTBN), denoted by S. The messages that are emitted by the parent nodes X_1 and X_2 are modelled as independent homogeneous continuous-time Markov processes $X_n(t)$, with state space $\mathcal{X}_n$ where $n \in \{1, 2\}$.

The agent node X_3 does not have direct access to the messages but observes a translation of them. The observation model is defined as the likelihood of a translation given the parent messages.

$$\psi(x_1, x_2) \coloneqq p(y(t) \mid X_1(t) = x_1, X_2(t) = x_2) \tag{2.1}$$

The agent X_3 is modelled as inhomogeneous continuous-time Markov process with state space $\mathcal{X}_3$ and set of actions $a \in A$ to choose from, where A is the action space.

Given the observation, the agent forms a belief over the parent states that summarizes the past observations. The belief state is denoted by $b(x_P; t)$, where x_P is joint parent states. The policy of the agent $\pi(a \mid b)$ is assumed to be shaped by evolution (close) to optimality. Based on the belief state, the agent takes an action, which in the setting described above corresponds to changing its internal dynamics.

2.2. Continuous-Time Bayesian Networks

Consider a directed acyclic graph denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges such that $\mathcal{E} = \{(m, n) : m, n \in \mathcal{V}\}$. In this graph, the parent nodes of node n are defined as the set of nodes that feed into it and denoted by $\text{Par}_{\mathcal{G}}(n) = \{m \in \mathcal{V} : (m, n) \in \mathcal{E}\}$. A directed acyclic graph is characterized as a Bayesian network where each node $n \in \mathcal{V}$ respresents a random variable X_n. This graph can be modelled as the joint distribution $p(X_1, X_2, ..., X_N)$ which factors as

$$p(X_1, X_2, ..., X_N) = \prod_{i=1}^{N} p(X_i \mid \text{Par}_{\mathcal{G}}(X_i)). \tag{2.2}$$

A continuous-time Bayesian network (CTBN) is a graphical model with a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that represents a collection of random variables whose values evolve continuously over time. In the CTBN framework, the dependencies of a set of Markov processes (MPs) can be modelled efficiently through a directed graph, relying on two assumptions. The first assumption is that only one node can transition at a time, and the second is that the instantaneous dynamics of each node depend only on its parent nodes [4, 18]. A two component CTBN is illustrated in Figure 2.2.

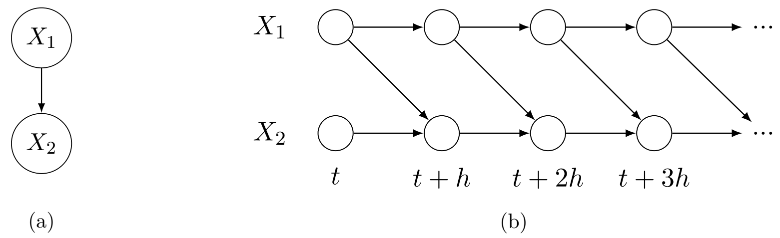

Figure 2.2.: Two component CTBN. (a) Network representation (b) CTBN in (a) unrolled in time as $h \to 0$

2.2.1. Continuous-Time Markov Processes

A continuous-time Markov process (CTMP) is a continuous-time stochastic process which satisfies the Markov property. In other words, given the current state, the probability distribution over the states in the future is conditionally independent of the past states [4].

Consider a CTMP $X(t)$ over a single variable with a countable state space $\mathcal{X}$. Then the Markov property can be written as

$$\Pr\left(X(t_l) = x_{t_l} \mid X(t_{l-1}) = x_{t_{l-1}}, \ldots, X(t_0) = x_{t_0}\right) = \Pr\left(X(t_l) = x_{t_l} \mid X(t_{l-1}) = x_{t_{l-1}}\right) \tag{2.3}$$

where $X(t)$ denotes the state of the variable at time t such that $X(t) = x_t \in \mathcal{X},\ t \geq 0$ and $t_0 < t_1 < ... < t_l$.

A CTMP is represented by its transition intensity matrix, $Q : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. In this matrix, the intensity q_i represents the intensity of leaving state x_i and $q_{i,j}$ represents the intensity of switching from state x_i to x_j, where $x_i, x_j \in \mathcal{X}$. The intensity matrix Q can be written in the following form:

$$Q = \begin{bmatrix} -q_1 & q_{1,2} & \cdots & q_{1,n} \\ q_{2,1} & -q_2 & \cdots & q_{2,n} \\ \vdots & \vdots & \ddots & \cdots \\ q_{n,1} & q_{n,2} & \cdots & -q_n \end{bmatrix} \tag{2.4}$$

where $q_i = \sum_{i \neq j} q_{i,j}$ [18].

2.2.1.1. Homogenous Continuous-Time Markov Processes

A continuous-time Markov process is time-homogenous when the transition intensities do not depend on time. Let X(t) be a homogenous CTMP, with a countable state space $\mathcal{X}$ and transition intensity matrix Q. Infinitesimal transition probability from state x_i to x_j in terms of the transition intensities $q_{i,j}$ can be written as [4]:

$$p_{i,j}(h) = \delta_{ij} + q_{i,j}h + o(h) \tag{2.5}$$

where $p_{i,j}(h) \equiv \Pr(X(t+h) = x_j \mid X(t) = x_i)$ are Markov transition functions, $\delta_{i,j} = \delta(x_i, x_j)$ is the Kronecker delta and $o(\cdot)$ is a function decaying to zero faster than its argument such that $\lim_{h\to 0} \frac{o(h)}{h} = 0$.

The *Chapman-Kolmogorov-* or *master-equation* is then derived as follows:

$$\begin{aligned} p_j(t) &\equiv \Pr(X(t) = x_j) \\ &= \sum_{\forall i} p_{i,j}(h) p_i(t-h) \\ \lim_{h\to 0} p_j(t) &= \lim_{h\to 0} \sum_{\forall i} \left[\delta_{ij} + q_{i,j}h + o(h)\right] p_i(t-h) \\ &= \lim_{h\to 0} p_j(t-h) + \lim_{h\to 0} h \sum_{\forall i} q_{i,j} p_i(t-h) \\ \lim_{h\to 0} \frac{p_j(t) - p_j(t-h)}{h} &= \lim_{h\to 0} \sum_{\forall i} q_{i,j} p_i(t-h) \\ \frac{d}{dt} p_j(t) &= \sum_{\forall i} q_{i,j} p_i(t) \end{aligned} \tag{2.6}$$

Equation 2.6 can be written in matrix form as

$$\frac{d}{dt} p(t) = p(t)Q \tag{2.7}$$

where the time-dependent probability distribution $p(t)$ is a row vector with entries $\{p_i(t)\}_{x_i \in \mathcal{X}}$. The solution of the system of ordinary differential equations (ODEs) is

$$p(t) = p(0)\exp(tQ) \tag{2.8}$$

with initial distribution $p(0)$.

The amount of time staying in a state x_i is exponentially distributed with parameter q_i. The probability density function f and the cumulative distribution function F for staying in the state x_i can be written as [18]

$$f(t) = q_i \exp\left(-q_i t\right), t \geq 0 \tag{2.9}$$

$$F(t) = 1 - \exp\left(-q_i t\right), t \geq 0. \tag{2.10}$$

Given the transition from state x_i, the probability of landing on state x_j is $q_{i,j}/q_i$.

Likelihood Function

Consider a single transition denoted as $d =< x_i, x_j, t >$, where the transition occurs from state x_i to x_j after spending t amount of time in state x_i. The likelihood of this transition is the product of the probability of having remained in state x_i for duration t from Equation 2.9, and the probability of transitioning to x_j.

$$p(d \mid Q) = (q_i \exp(-q_i t)) \left(\frac{q_{i,j}}{q_i}\right) \tag{2.11}$$

The likelihood of a trajectory sampled in time interval $[0, T]$ from a homogenous CTMC, denoted by $X^{[0,T]}$, can be decomposed as the product of the likelihood of single transitions. The summary statistics over this trajectory can be written as $\Upsilon(x_i)$, the total amount of time spent in state x_i, and $r(x_i, x_j)$, the total number of transitions from state x_i to x_j [19]. These summary statistics can be formulated as

$$\Upsilon(x_i) = \sum_{d \in X^{[0,T]}} \delta(X(t), x_i) \tag{2.12}$$

$$r(x_i, x_j) = \sum_{d \in X^{[0,T]}} \delta(X(t), x_i)\delta(X(t+h), x_j) \tag{2.13}$$

where δ is the Kronecker delta. Then the likelihood of a trajectory $X^{[0,T]}$ can be written as:

$$\begin{aligned} p(X^{[0,T]} \mid Q) &= \prod_{d \in X^{[0,T]}} p(d \mid Q) \\ &= \left(\prod_i q_i^{r(x_i)} \exp\left(-q_i \Upsilon(x_i)\right)\right) \left(\prod_i \prod_{j \neq i} \left(\frac{q_{i,j}}{q_i}\right)^{r(x_i,x_j)}\right) \\ &= \prod_{j \neq i} \exp(-q_{i,j} \Upsilon(x_i)) \; q_{i,j}^{r(x_i,x_j)} \end{aligned} \tag{2.14}$$

where $r(x_i) = \sum_{j \neq i} r(x_i, x_j)$ is the total number transitions leaving state x_i.

2.2.1.2. Conditional Markov Processes

A CTMP is *time-inhomogenous* when the transition intensities change over time. In a CTBN, while every node is a Markov process, the leaf nodes are characterized as *conditional* Markov processes, a type of inhomogeneous MP, where the intensities change over time, but not as a function of time rather as a function of parent states [18].

Let X be a conditional MP in a graph $\mathcal{G}$, with a set of parents $U = \mathrm{Par}_{\mathcal{G}}(X)$. Its *conditional intensity matrix* (CIM) $Q_{X\,|\,U}$ can be viewed as a set of homogenous intensity matrices $Q_{X\,|\,u}$, with entries $q_{i,j|u}$ (similar to Equation 2.4), for each instantiation of parent nodes $U(t) = u$ such that $u \in \mathcal{U} = \bigtimes_{X_m \in \mathrm{Par}_{\mathcal{G}(X)}} \mathcal{X}_m$, where $\bigtimes$ denotes Cartesian product [18]. As a result, given a trajectory of parent nodes, X has a trajectory of intensity matrix as

$$Q^{[0,T]} = [Q_{X\,|\,U(t_0)}, Q_{X\,|\,U(t_1)}, ..., Q_{X\,|\,U(t_N)}],\ 0 < t_0 < ... < t_N \leq T. \tag{2.15}$$

Markov transition function for a conditional MP can be written as

$$\Pr(X(t+h) = x_j \mid X(t) = x_i, U(t) = u, Q_{X\,|\,u}) = \delta(i,j) + q_{i,j|u}h + o(h). \tag{2.16}$$

Likelihood Function

Given the instantiation of its parents, the complete information on the dynamics of X is obtained. Then the likelihood of a trajectory drawn from a conditional MP X can be written similar to Equation 2.14,

$$\begin{aligned} p(X^{[0,T]} \mid Q_{X\,|\,U}) &= \left(\prod_u \prod_i q_{i|u}^{r(x_i|u)} \exp\left(-q_{i|u}\Upsilon(x_i \mid u)\right)\right) \left(\prod_u \prod_i \prod_{j\neq i} \left(\frac{q_{i,j|u}}{q_{i|u}}\right)^{r(x_i,x_j|u)}\right) \\ &= \prod_u \prod_{j\neq i} \exp(-q_{i,j|u}\Upsilon(x_i \mid u))\ q_{i,j|u}^{r(x_i,x_j|u)} \end{aligned} \tag{2.17}$$

where the summary statistics $\Upsilon(\cdot)$ and $r(\cdot)$, introduced in Equations (2.12)-(2.13), are also conditioned on parent nodes.

2.2.2. The CTBN Model

Evidently, a homogeneous CTMP can be considered as a conditional MP whose set of parents is empty. Thus, a CTBN can be formed as a set of conditional Markov processes.

Let S be a CTBN with graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and nodes $\mathcal{V} = \{1, ..., N\}$. Each node represents a local variable $X_n,\ n \in \mathcal{V}$, with a countable state space $\mathcal{X}_n$. This results in factorizing state spaces for S such that $\mathcal{S} = \mathcal{X}_1 \times \mathcal{X}_2 \times ... \times \mathcal{X}_N$. The joint states of the variables are denoted by $s = (x_1, x_2, ..., x_N) \in \mathcal{S}$ where $x_1 \in \mathcal{X}_1, ..., x_N \in \mathcal{X}_N$. The dependencies of each variable are defined as a set of its parents $U_n = \mathrm{Par}_{\mathcal{G}}(X_n)$ with values $U_n(t) = u_n$ such that $u_n \in \mathcal{U}_n = \bigtimes_{X_m \in \mathrm{Par}_{\mathcal{G}(X_n)}} \mathcal{X}_m$. In the following, the set of all conditional transition intensity matrices are denoted as $\boldsymbol{Q} = \left\{Q_{X_1\,|\,U_1}, ..., Q_{X_N\,|\,U_N}\right\}$.

Consider a trajectory drawn from S, such that $S^{[0,T]} = \left\{X_1^{[0,T]}, X_2^{[0,T]}, ..., X_N^{[0,T]}\right\}$. Following Equation 2.17, the likelihood of this trajectory can be written as

$$p(S \mid \boldsymbol{Q}) = \prod_{n=1}^{N} \prod_{u \in \mathcal{U}_n} \prod_{x_i \in \mathcal{X}_n} \prod_{x_j \in \mathcal{X}_n \setminus x_i} \exp\left(q_{i,j|u}^n \Upsilon_n(x_i \mid u)\right) (q_{i,j|u}^n)^{r_n(x_i,x_j|u)}. \tag{2.18}$$

It should be noted that a CTBN can also be represented by a single CIM, through *amalgamation* operation [18].

2.3. Partially Observable Markov Decision Processes

The partially observable Markov decision process (POMDP) framework provides a model of an agent which interacts with its environment but is unable to obtain certain information about its state. Instead, the agent receives an observation which is a function of the true state, e.g. noisy observations, translations, etc. The main goal, similar to Markov decision processes (MDPs), is to learn a policy performing a task by optimizing a reward function.

The problem of decision making under uncertainty can be considered in two parts. The first is to keep a belief state which summarizes past experiences, and the second is to optimize a policy π which will select an action based on the belief state [12, 16].

Consider a POMDP represented as a tuple $(\mathbb{S}, A, T, R, \mathcal{Y}, \psi)$, where $\mathbb{S}$ is a countable set of states of the world, A is a set of actions, $T : \mathbb{S} \times A \rightarrow \Pi(\mathbb{S})$ is the state-transition function, $R : \mathbb{S} \times A \rightarrow \mathbb{R}$ is reward function, $\mathcal{Y}$ is set of observations and $\psi : \mathbb{S} \times A \rightarrow \Pi(\mathcal{Y})$ is the observation function. The transition function T gives a probability distribution over the next states s', given a state and action, such that $T(s, a, s') = \Pr(s' \mid s, a)$ and $s, s' \in \mathbb{S}$, $a \in A$. The observation function ψ gives a probability distribution over observations given a state and action, such that $\psi(s', a, y) = \Pr(y \mid s', a)$ and $y \in \mathcal{Y}$. The reward function R gives the expected immediate reward for each action and state $R(s, a)$.

The belief state, if represented as a probability distribution over states, provides a summary over the agent's past experiences. This representation also allows the agent to account for its uncertainty while making decisions. The optimal policy of a POMDP agent leads to optimal behaviour as a function of the agent's belief state. The expected amount of future rewards upon executing a policy is defined as *value function* and used to evaluate a given policy.

Finite-horizon model represents a problem where the agent has t steps to take, that is given the current state of the world, the agent will receive an observation and make a decision t times. In this model, it is more likely that the agent has a nonstationary policy, a policy which is a sequence of policies indexed by time. This policy corresponds to a time-dependent behaviour. Consider the famous and well-studied tiger problem as an example [12, 17]. In this problem, the agent is in front of two doors, one of which has a tiger behind, the other has a reward. The agent has three actions to choose from, it can open the left door or the right door, or it can listen to obtain information. If the agent can only take one step, it might opt for opening one of the doors even though there is uncertainty about the location of the tiger. However, when there are two steps to go, listening for more information in the first step to decrease the uncertainty, would be a better choice.

A nonstationary t-step policy can be represented as a *policy tree* ρ shown in Figure 2.3. The policy tree describes the optimal behaviour for t steps, conditioned on the observation. The top node shows the first action to be taken, then depending on the observation, a different branch is followed until the end of the steps.

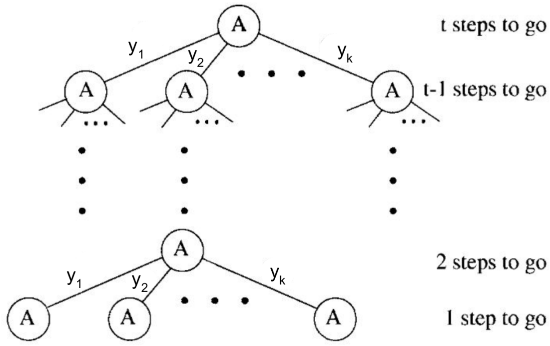

Figure 2.3.: A policy tree of depth t representing t-step nonstationary policy. Source: [12]

The value of a nonstationary t-step tree ρ is denoted by $V_\rho(s)$ and given in Equations (2.19)-(2.21), where s is starting state, $a(\rho)$ is the first action at the top of the tree, s' is the next state. $y_i(\rho)$ denotes the (t-1)-step policy that has been chosen after taking the action $a(\rho)$ and getting observation y_i [12]. γ is the *discount factor*, such that $0 \leq \gamma \leq 1$, used to regulate the contribution of the future rewards to the value function [28].

$$\begin{aligned} V_\rho(s) &= R(s, a(\rho)) + \gamma \cdot (\text{ Expected value of the future }) && (2.19)\\ &= R(s, a(\rho)) + \gamma \sum_{s' \in \mathbb{S}} \Pr\left(s' \mid s, a(\rho)\right) \sum_{y_i \in \mathcal{Y}} \Pr\left(y_i \mid s', a(\rho)\right) V_{y_i(\rho)}\left(s'\right) && (2.20)\\ &= R(s, a(\rho)) + \gamma \sum_{s' \in \mathbb{S}} T\left(s, a(\rho), s'\right) \sum_{y_i \in \mathcal{Y}} \psi\left(s', a(\rho), y_i\right) V_{y_i(\rho)}\left(s'\right) && (2.21) \end{aligned}$$

The value of a policy tree ρ starting from state s has two components, as can be seen in Equation 2.19. The first component is the immediate reward, that is, the reward that the agent will get for taking action $a(\rho)$ while in state s. The second component is the expected future reward, discounted by γ. This value is computed by taking the expectation over next possible states $s' \in \mathcal{S}$, and their values. The value of a state depends on the observation that the agent receives, which will determine the (t-1)-step policy to be executed. Therefore, a second expectation is performed over the observations.

As mentioned before, the agent is never certain about the state of the world. Therefore, the relevant value function is that of a policy tree starting from a belief state b, and calculated as the expected value over the states.

$$V_\rho(b) = \sum_{s \in \mathcal{S}} b(s) V_\rho(s) \tag{2.22}$$

Equation 2.22 gives the value of a policy tree starting from belief state b. The optimal policy is then chosen as the one which has the maximum value. The optimal t-step value starting from belief state b is executing the best policy tree for that belief state.

$$V_t(b) = \max_{\rho \in \mathcal{P}} V_\rho(b) \tag{2.23}$$

where $\mathcal{P}$ denotes a finite set of policy trees. It is noteworthy that the value function of every policy tree V_ρ is linear in b. As can be seen from Equation 2.23, V_t is defined as the maximum of all V_ρ over b. It is the envelope of these value functions; therefore, it is piecewise-linear and convex.

The infinite-horizon discounted model considers the value function over an infinitely long trajectory of the agent. In a POMDP problem, the infinite horizon discounted value function is still convex [32], but it may not always be piecewise-linear. However, it can be approximated by a finite horizon value function for sufficiently many steps [24, 26].

In the problem considered in this thesis, the agent node X_3 in Figure 2.1 cannot observe the incoming messages directly, rather a summary of them. This setting presents a POMDP problem. However, since the optimal policy of the agent is assumed to be given, the main focus in the POMDP framework is belief state estimation. In the following, update methods for the belief state are introduced, where belief state refers to the posterior probability distribution over the environment states.

2.3.1. Exact Belief State Update

In a scenario where compact representations of the transition function $T(s, a, s')$ and observation function $\psi(s', a, y)$ are available, the belief state update can be obtained as [12]

$$\begin{aligned} b'(s') &= \Pr(s' \mid y, a, b) && (2.24)\\ &= \frac{\Pr(y \mid s', a, b)\Pr(s' \mid a, b)}{\Pr(y \mid a, b)} && (2.25)\\ &= \frac{\Pr(y \mid s', a)\sum_{s \in \mathcal{S}} \Pr(s' \mid a, b, s)\Pr(s \mid a, b)}{\Pr(y \mid a, b)} && (2.26)\\ &= \frac{\psi(s', a, y)\sum_{s \in \mathcal{S}} T(s, a, s')\, b(s)}{\Pr(y \mid a, b)}. && (2.27) \end{aligned}$$

Equations (2.24)-(2.25) are obtained following Bayes' theorem, and Equations (2.25)-(2.26) follow from the law of total probability.

From Equation 2.8, the relation between transition model T and transition intensity matrix Q can be written as $T = \exp(tQ)$. It is noteworthy that the denominator in Equation 2.27 is in the following form,

$$\Pr(y \mid a, b) = \sum_{\forall s' \in \mathcal{S}} \psi(y \mid s', a) \sum_{\forall s \in \mathcal{S}} T(s' \mid s, a) b(s) \tag{2.28}$$

which is computationally expensive in the case of continuous state space.

2.3.2. Exact Belief State Update for CTMP by Filtering

Equation 2.27 is the discrete-time solution of belief state. However, since in the model described in Section 2.1, the parent nodes are modelled as CTMPs, the environment state evolves as the state of a CTMP. Therefore, the belief over the environment states should be solved in continuous-time. This is achieved by the inference of the posterior probability of CTMP [11].

A *filtering problem* in statistical context refers to the inference of the conditional probability of the current state of the system at a given point in time, given the past observations [10].

Let X be a CTMP with transition intensity matrix Q. Assume discrete-time observations denoted by $y_1 = y(t_1), ..., y_L = y(t_L)$. The belief state can be written as:

$$b(x_i; t_L) = \Pr(X(t_L) = x_i \mid y_1, ..., y_L) \tag{2.29}$$

From the master equation given in Equation 2.6, it follows that:

$$\frac{d}{dt} b(x_j; t) = \sum_{\forall i} q_{i,j} \cdot b(x_i; t) \tag{2.30}$$

The time-dependent belief state $b(t)$ is a row vector with $\{b(x_i; t)\}_{x_i \in \mathcal{X}}$. This posterior probability can be described by a system of ODEs:

$$\frac{db(t)}{dt} = b(t)Q \tag{2.31}$$

where the initial condition $b(0)$ is a row vector with $\{b(x_i; t)\}_{x_i \in \mathcal{X}}$ [11]. The solution is

$$b(t) = b(0) \exp(tQ). \tag{2.32}$$

The belief state update at discrete times of observation y_t is derived as

$$\begin{aligned}
b(x_i; t_L) &= \Pr(X(t_L) = x_i, \mid y_1, ..., y_L) \\
&= \frac{\Pr(y_1, ..., y_L, X(t_L) = x_i)}{\Pr(y_1, ..., y_L)} \\
&= \frac{\Pr(y_L \mid y_1, ..., y_{L-1}, X(t_L) = x_i)}{\Pr(y_L \mid y_1, ..., y_{L-1})} \frac{\Pr(y_1, ..., y_{L-1}, X(t_L) = x_i)}{\Pr(y_1, ..., y_{L-1})} \\
&= Z_L^{-1} \ \Pr(y_L \mid X(t_L) = x_i) \ \Pr(X(t_L) = x_i \mid y_1, ..., y_{L-1}) \\
&= Z_L^{-1} \ p(y_L \mid x_i) \ b(x_i; t_L^-)
\end{aligned} \tag{2.33}$$

where $Z_L = \sum_{x_i \in \mathcal{X}} p(y_L \mid x_i) \ b(x_i; t_L^-)$ is the normalization factor [11].

2.3.3. Belief State Update using Particle Filter

In a more realistic scenario, the exact update of belief state may not be feasible for several reasons. The computation of exact belief update is expensive for large state spaces, which can be seen from Equation 2.28. Moreover, a problem with continuous state spaces requires a belief state represented as probability distributions over an infinite state space rather than a collection of probabilities as given in Section 2.3.1 [30]. Such representation cannot be obtained using the exact method. Another reason is the lack of a compact representation of transition or observation models. Under such circumstances, the belief state is obtained using sample-based approximation methods [30].

It should be noted that since the belief state is nothing but the conditional probability of true states given the observations, the problem at hand poses a filtering problem as described in Section 2.3.2.

2.3.3.1. Particle Filtering

Particle filtering is one of the most commonly used Sequential Monte Carlo (SMC) algorithms. The popularity of this method thrives from the fact that, unlike other approximation methods such as Extended Kalman Filter, it does not rely on linearization approximations or assume Gaussian distribution. This advantage offers great flexibility and finds application in a wide range of areas [5].

The key idea in particle filtering is to approximate a target distribution $p(x)$ by a set of samples, i.e. particles, drawn from that distribution. This is achieved by sequentially updating the particles through two steps. The first step is *importance sampling.* Since the target distribution is not available, the particles are generated from a *proposal distribution* $q(x)$ and weighted according to the difference between target and proposal distributions. The second step is to resample the particles using these weights with replacement [10].

Consider a problem of deriving the expectation $\hat{f}(x) = \mathbb{E}_{p(x)}\left[f(x)\right] = \int f(x)p(x)dx$, and suppose $p(x)$ is an intractable density function from which the particles cannot be sampled. Instead they are drawn from a proposal distribution $q(x)$, which yields an empirical approximation such that

$$
\begin{aligned}
x^{(m)} &\sim q(x) \\
q(x) &\approx \frac{1}{M}\sum_{m=1}^{M}\delta_{x^{(m)}}(x)
\end{aligned}
$$

where $\delta_{x^{(m)}}(x)$ is Dirac delta and M is the number of particles. The expectation in question can be written as

$$
\begin{aligned}
\int f(x)p(x)dx &= \int f(x)\frac{p(x)}{q(x)}q(x)dx \\
\int f(x)\frac{p(x)}{q(x)}\left(\frac{1}{M}\sum_{m=1}^{M}\delta_{x^{(m)}}(x)\right)dx &= \frac{1}{M}\sum_{m=1}^{M}\frac{p(x^{(m)})}{q(x^{(m)})}f(x^{(m)})
\end{aligned}
$$

where $w(x^{(m)}) = \frac{p(x^{(m)})}{q(x^{(m)})}$ is defined as *importance weight* of a particle. Then the particles are resampled using the importance weights with replacement, which concludes one iteration of sequential updating [10].

2.3.3.2. Marginalized Continuous-Time Markov Processes

In this work, the particles to represent the belief state are drawn from a marginalized CTBN. Consider a CTBN, denoted by S, with local variables X_n, $n \in \{1, ..., N\}$, and a set of CIMs $\boldsymbol{Q}$. In the following, it is assumed that every non-diagonal entry in $Q_{n|u}$ is Gamma distributed with shape and rate parameters, $\alpha^n_{i,j|u}$ and $\beta^n_{i,j|u}$.

The marginal process description of S considering a single trajectory in the interval $[0, t)$ can be written as

$$\Pr(X_n(t+h) = x_j \mid X_n(t) = x_i, U_n(t) = u, S^{[0,t)}) \tag{2.34}$$

$$= \int \Pr(X_n(t+h) = x_j \mid X_n(t) = x_i, U_n(t) = u, Q_{n|u}, S^{[0,t)}) p(Q_{n|u}) dQ_{n|u} \tag{2.35}$$

$$= \delta_{i,j} + \mathbb{E}[q^n_{i,j|u} \mid S^{[0,t]} = s^{[0,t]}]\ h + o(h). \tag{2.36}$$

By integrating out the intensity matrix $Q_{n|u}$, the parameter is replaced by its expected value given the history of the process. It should be noted that by doing so, the process becomes parameter-free, and thus self-exciting [27].

The derivation of the conditional expectation for a marginal CTBN follows from the Bayes' rule:

$$p\left(\boldsymbol{Q} \mid S^{[0,t]}\right) = \frac{p\left(S^{[0,t]} \mid \boldsymbol{Q}\right) p(\boldsymbol{Q})}{p\left(S^{[0,t]}\right)} \tag{2.37}$$

Equation 2.37, written for single trajectory $S^{[0,t]}$, can be extended for multiple trajectories. Consider K trajectories drawn from S, denoted by $\xi^K_t = \left\{S^{[0,t],1}, S^{[0,t],2}, ..., S^{[0,t],K}\right\}$. Since the trajectories are conditionally independent, given $\boldsymbol{Q}$, using Equation 2.18 the likelihood of set ξ^K_t is written as,

$$\Pr(\xi^K_t \mid \boldsymbol{Q}) = \prod_{n=1}^{N} \prod_{u \in \mathcal{U}_n} \prod_{x_i \in \mathcal{X}_n} \prod_{x_j \in \mathcal{X}_n \setminus x_i} \exp\left(q^n_{i,j|u} \Upsilon_n(x_i \mid u)\right) (q^n_{i,j|u})^{r_n(x_i, x_j|u)} \tag{2.38}$$

where the joint summary statistics of X_n over all K trajectories are denoted by $\Upsilon_n(x_i \mid u) = \sum_{\kappa=1}^{K} \Upsilon^\kappa_n(x_i \mid u)$ and $r_n(x_i, x_j \mid u) = \sum_{\kappa=1}^{K} r^\kappa_n(x_i, x_j \mid u)$.

Given independent Gamma-priors on transition intensities, the expectation in Equation 2.36 can be evaluated as follows:

$$\mathbb{E}\left[q^n_{i,j|u} \mid \xi^K_t\right] = \frac{\alpha^n_{i,j|u} + r_n(x_i, x_j \mid u)}{\beta^n_{i,j|u} + \Upsilon_n(x_i \mid u)} \tag{2.39}$$

The trajectories sampled from a CTBN are conditionally independent given the intensity matrices. However, the expectation of these matrices are used to sample from the marginalized CTBN. This expectation, evaluated using Equation 2.39, depends on the summary statistics $\Upsilon_n(x_i \mid u)$ and $r_n(x_i, x_j \mid u)$ from all observed trajectories up to the time of the evaluation. Therefore, in order to draw K trajectories from a marginalised CTBN exactly, they have to be simulated jointly. Since this simulation is computationally infeasible, we approximated this simulation where the particles are processed one after another. The approximation can be validated rigorously using collapsed variational inference but this derivation is beyond the scope of this thesis. Our results show this validity empirically. The algorithm for belief state update through marginal particle filtering is given in Chapter 3.

2.4. Sampling Algorithms

2.4.1. Gillespie Algorithm for Generative CTBN

Gillespie algorithm is a computer-oriented Monte Carlo simulation procedure that is originally proposed to simulate the reactions of molecules in any spatially homogeneous chemical system. Such systems are regarded as Markov processes and represented via their master equations, which cannot be directly used to obtain realizations of the process. Gillespie algorithm is an efficient tool to overcome this problem [9].
This algorithm can also be applied to sample *events* from a CTBN given the transition intensity matrices, where an event refers to a transition occurring at a specific point in time. This procedure is introduced as *Generative CTBN* in [18].

Algorithm 1: Generative CTBN

Input : Structure of the network with N local variables $X_1, X_2, ..., X_n$ with state-space $\mathcal{X}_n = \{x_1, ..., x_m\}$
Transition intensity matrices Q_n with entries $q^n_{i,j}$
T_{max} to terminate simulation
Output : Sample trajectory of the network
Initialize: Initialize node values $X_n(0) = x_i \in \mathcal{X}_n$

```
1: while t < T_max do
2:     τ ~ exp(Σ_{∀n} Σ_{∀i≠j} q^n_{i,j})
3:     transitioning node is randomly drawn with probability P(X_n) = q^n_i / Σ_{∀n} q^n_i
4:     next state is randomly drawn with probability P(x_j) = q^n_{i,j} / q^n_i
5:     t ← t + τ
6: end while
```

2.4.2. Thinning Algorithm

Thinning algorithm is a method introduced to simulate nonhomogenous Poisson processes [14]. Later, it is adapted to sample from Hawkes process, a self-exciting process with time-dependent intensity function [20, 23]. This algorithm is used here to simulate the inhomogeneous Markov process.

Algorithm 2: Thinning Algorithm

Input : $\lambda(t)$ the intensity function of the inhomogenous process
N number of events to terminate simulation

Output : Sample trajectory of the process

Initialize: Time $t = 0$

```
while i < N do
    the upper bound for intensity, λ*
    transition time τ drawn by  u ~ U(0,1)  and  τ = −ln(u)/λ*
    t ← t + τ
    draw  s ~ U(0,1)
    if  s ≤ λ(t)/λ*  then
        sample accepted and  t_i = t, i = i + 1
    end if
end while
```

3. Methodology

This chapter presents the methodology used in this thesis. First, it is explained how different frameworks, which were introduced in Chapter 2, are put into use. Then, the algorithms used in data generation and inference are given in detail. The results from these experiments are presented in Chapter 4.

3.1. The Model

A detailed graphical model explored in this thesis is given in the Figure 3.1. This model presents an intersection of CTBN and POMDP frameworks.

- The transition models of the nodes X_1, X_2 and X_3, and the dependencies between them are modelled as CTBN.
- The interaction of agent node X_3 and its environment is modelled as POMDP.

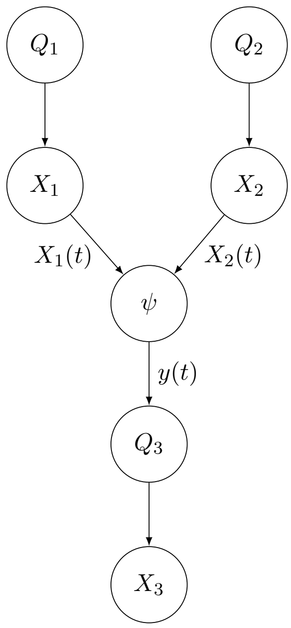

Figure 3.1.: Hierarchical model.

3.1.1. CTBN Model

The transition models of the nodes and the dependencies between them are modelled as CTBN, denoted by S, with graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, 3\}$ and $\mathcal{E} = \{(1, 3), (2, 3)\}$. The state of the nodes evolve as local variables X_n, $n \in \mathcal{V}$, and each variable has a state space denoted by $\mathcal{X}_n$. The network S represents a stohastic process over a structured factorising state space $\mathcal{S} = \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{X}_3$.

The parent nodes X_1 and X_2 emit their states as messages.
The dynamics of these nodes are modelled as independent homogeneous continuous-time Markov processes $X_n(t)$, with binary-valued states $\mathcal{X}_n = \{0, 1\}$ for $n \in \{1, 2\}$. These processes are defined by transition intensity matrices Q_n, which are assumed to be Gamma distributed with shape and rate parameters $\boldsymbol{\alpha}^n = [\alpha_0^n, \alpha_1^n]$ and $\boldsymbol{\beta}^n = [\beta_0^n, \beta_1^n]$, respectively, and are in the

following forms.

$$Q_n = \begin{bmatrix} -q_0^n & q_0^n \\ q_1^n & -q_1^n \end{bmatrix} \tag{3.1}$$

$$Q_n \sim \mathrm{Gam}(\boldsymbol{\alpha}^n, \boldsymbol{\beta}^n) \quad \text{for } n \in \{1, 2\} \tag{3.2}$$

It should be noted that in Equation 3.1, the suffixes are simplified using the fact that $q_i = \sum_{i \neq j} q_{i,j}$.

The agent X_3 is modelled as inhomogenouos continuous-time Markov process with binary states $\mathcal{X}_3 = \{0, 1\}$, a set of actions $a \in \{a_0, a_1\}$, and a set of transition intensity matrices which contains one matrix corresponding to each action, $\boldsymbol{Q}_3 = \left\{Q_{3|a_0}, Q_{3|a_1}\right\}$.

The dependencies are represented by a set of parents for each node $U_n = \mathrm{Par}_{\mathcal{G}}(X_n)$ and for the model shown in Figure 3.1 can be written as follows:

$$U_1, U_2 = \emptyset$$
$$U_3 = \{X_1, X_2\}$$

In order to have a compact representation of parent messages, a subsystem of S consisting of only the parent nodes X_1 and X_2 can be considered as a single system. These two processes can be represented as a *joint* process, X_P, with factorising state space $\mathcal{X}_P = \mathcal{X}_1 \times \mathcal{X}_2$. The transition intensity matrix of the new joint system, Q_P is obtained by amalgamation operation, denoted by *, between Q_1 and Q_2 (see Appendix A) [18].

$$Q_P = Q_1 * Q_2 \tag{3.3}$$

3.1.2. POMDP Model

In a conventional POMDP scenario, there are two problems to be addressed, one is belief state update and the other is policy optimization. As mentioned in Section 2.3, in the problem at hand, the policy of agent X_3 is assumed to be optimal and given. Thus, the POMDP model of the agent only consists of belief state update. A detailed view of the agents interaction with its environment from POMDP framework perspective is given in the Figure 3.2.

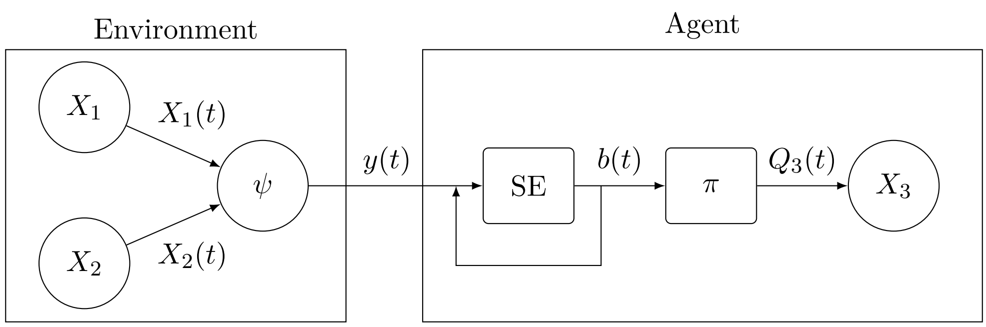

Figure 3.2.: A closer look at the agent-environment interaction from the perspective of POMDP framework.

It should be noted that, the interaction in Figure 3.2 is only one-sided, the state or action of the agent does not affect the environment.

3.1.2.1. Observation Model

The messages sent by the parent nodes are translated by the observation model. The agent node X_3 does not have a direct access to the messages, but observes a translation of them. The observation is denoted by $y(t) = y_t$ such that $y_t \in \mathcal{Y}$ where $\mathcal{Y}$ is the observation space. The observation model defines a probability distribution over the observation for each combination of parent messages.

$$\psi(x_1, x_2) = p(y(t) \mid X_1(t) = x_1, X_2(t) = x_2) \tag{3.4}$$

where $x_1 \in \mathcal{X}_1$ and $x_2 \in \mathcal{X}_2$. As explained in Section 3.1.1, using the joint process X_P for the sake of conciseness, Equation 3.4 can be written as

$$\psi(x_P) = p(y(t) \mid X_P(t) = x_P) \tag{3.5}$$

where $x_P \in \mathcal{X}_P$.

$\psi(x_P)$ is defined as deterministic categorical distribution over the observation space $\mathcal{Y}$. For each state x_P, there is one possible observation $y_i \in \mathcal{Y}$, such that

$$p(y_i \mid x_P) = 1 \ \wedge \ p(y_j \mid x_P) = 0 \quad \forall j \neq i. \tag{3.6}$$

ψ denotes the matrix with rows $\{\psi(x_P)\}_{x_P \in \mathcal{X}_P}$.

3.1.2.2. Belief State

The belief state provides a summary over agent's past experiences and allows the agent to take its own uncertainty into account. The belief state is formed by the *state estimator* (labelled as *SE* in Figure 3.2) over the parent states, denoted by $b(x_P; t)$.

$$b(x_P; t) = \Pr(X_P(t) = x_P \mid y_1, ..., y_t) \tag{3.7}$$

The initial belief state is assumed to be uniformly distributed, and is updated with the first observation at $t_0 = 0$.

Exact Belief State Update

As discussed in Section 2.3.2, given the transition intensity matrices of parent nodes, Q_1 and Q_2, the continuous-time belief state update poses a filtering problem for CTMPs. This problem can be formulated according to the joint process of parents.

$$b(x_P; t) = \Pr(X_P(t) = x_p \mid y_1, ..., y_t) \tag{3.8}$$

Consider discrete-time observations from this process, denoted by $y_1 = y(t_1), ..., y_l = y(t_l)$ and time-dependent belief state $b(t)$ as a row vector with $\{b(x_P; t)\}_{x_P \in \mathcal{X}_P}$. Following Equation 2.32 and Equation 2.33, the belief state update is evaluated as

$$b(t) = b(0) \exp(tQ_P) \tag{3.9}$$

with the initial condition $b(0)$. The update at discrete times of observation y_l is

$$\begin{aligned} b(x_P; t_l) &= Z_l^{-1}\ p(y_l \mid X_P(t_l) = x_P)\ b(x_P; t_l^-) & (3.10) \\ &= Z_l^{-1}\ \psi(x_P)\ b(x_P; t_l^-) & (3.11) \end{aligned}$$

where $Z_l = \sum_{x_P \in \mathcal{X}_P} \psi(x_P)\ b(x_P; t_l^-)$ is the normalization factor.

Belief State Update Using Marginalized Particle Filter

The assumption that the complete information of parent dynamics is available is unrealistic. In an environment as described above, the agent is more likely not to have access to the parameters Q_1 and Q_2. It may rather have some prior beliefs over them. Moreover, when the state estimator utilizes exact update method, these parameters are assumed to be available for the inference as well. Thus, in order to simulate a more realistic model and be able to marginalize out these parameters from inference problem, the joint parent process X_P is replaced with its marginalized counterpart. Using the Gamma-priors over Q_1 and Q_2 (Equation 3.2) and sufficient statistics over the particle history, the particles are drawn from this marginalized process as explained in Section 2.3.3.2. With every new observation, the particles are propogated through the marginal process. The processing of the particles are done one after another as explained in Section 2.3.3.2. After propagating each particle, the summary statistics are updated and the parameters are re-estimated using the Equation 2.39. The belief state is then obtained as the distribution of states over the particles,

$$b(x_P; t) = \frac{1}{M} \sum_{m=1}^{M} \delta_{k_m(t), x_P} \tag{3.12}$$

where M is the number of particles, $k_i \in \mathbf{k}$ is the set of particles, and δ is the Kronecker delta.

In Algorithm 3, the weight update for the particles is performed on line 4, based on the observation model. Given the deterministic nature of the observation model as described in Equation 3.6, in rare cases where the observation y_k stems from an unlikely transition of parent nodes, the particles fail to simulate this transition and all of them are rejected. One possible solution to this problem would be to increase the number of particle to increase the probability of sampling unlikely transitions. However, in practice, this solution is computationally infeasible. Instead, the degeneration of the particle filter is dealt with by assigning uniform probabilities to the particles, effectively ignoring the unlikely transitions. The situation is illustrated by examples from simulation in Section 4.2.

Algorithm 3: Marginal particle filter for belief state update [27]

Input : Observation y_l at time t_l, set of particles $\mathbf{k}^{l-1}$, estimated $\hat{Q}$
Output: New set of particles $\mathbf{k}^l$, $\mathbf{b}^{[t_{l-1},t_l]}$

```
1: for k_m ∈ k^{l-1} do
2:   k_m = {x_m, Q̂} ← Propagate particle through marginal process from t_{l-1} to t_l
3:   Q̂ ← sufficient statistics added from k_m[t_{l-1}, t_l]
     // observation likelihood assigned as particle weight
4:   w_m ← p(y_l | X_P(t_l) = x_m)
5: end for
   // belief state from t_{l-1} to t_l
6: b^{[t_{l-1},t_l]} ← { (1/M) Σ_{m=1}^{M} δ_{k_m^{[t_{l-1},t_l]}, x_P} }_{x_P ∈ X_P}
   // normalize weights
7: w_m ← w_m / Σ_m w_m
   // resample particles
8: for k_m ∈ k^l do
9:   k_m ← Sample from k^l with probabilities w_m with replacement
10: end for
```

Step 2: $k_m = \left\{x_m, \hat{Q}\right\} \leftarrow$ *Propagate particle through marginal process from* t_{l-1} *to* t_l

Step 6: $\mathbf{b}^{[t_{l-1},t_l]} \leftarrow \left\{\frac{1}{M}\sum_{m=1}^{M}\delta_{k_m^{[t_{l-1},t_l]},x_P}\right\}_{x_P\in\mathcal{X}_P}$

Step 7: $w_m \leftarrow \frac{w_m}{\sum_m w_m}$

3.1.2.3. Optimal Policy

The optimal policy is defined using a polynomial function of belief state.

$$\pi(b) = \begin{cases} a_0 & \text{if } wb^\intercal > 0.5 \\ a_1 & \text{otherwise} \end{cases} \tag{3.13}$$

where w is a row vector of weights.

Given the optimal policy, $\pi(b)$, the agent takes an action based on the belief state. In the setting described above, taking an action means to change its internal dynamics to the transition intensity matrix corresponding to that action.

$$a(t) = \pi(b(t)) \tag{3.14}$$

$$Q_3(t) = \begin{cases} Q_{3|a_0} & \text{if } a(t) = a_0 \\ Q_{3|a_1} & \text{otherwise} \end{cases} \tag{3.15}$$

3.2. Inference of Observation Model

Inference problem is considered for deterministic observation models, such that each state $x_P \in \mathcal{X}_P$ can only be translated to one observation. Considering the number of states of parents and the observations, this results in a number of possible observation models.

Consider a trajectory in the dataset, denoted by $S^{[0,T]} = \left\{X_1^{[0,T]}, X_2^{[0,T]}, X_3^{[0,T]}\right\}$. The set of parameters to the system, as introduced before, is written as $\theta = \{Q_1, Q_2, \boldsymbol{Q}_3, \pi, \psi\}$. Given the parent trajectories $X_1^{[0,T]}$ and $X_2^{[0,T]}$, the belief state and the resulting intensity matrix trajectory $Q_3^{[0,T]}$ are computed for each observation model. Then the likelihood of sample trajectory $S^{[0,T]}$ given the parameters θ are used for classifiation task.

3.2.1. Likelihood Model

The likelihood of a sample trajectory $S^{[0,T]}$ can be written as:

$$\begin{aligned}
p(S^{[0,T]} \mid \theta) &= p(X_1^{[0,T]}, X_2^{[0,T]}, X_3^{[0,T]} \mid Q_1, Q_2, \boldsymbol{Q}_3, \pi, \psi) \\
&= p(X_3^{[0,T]} \mid X_1^{[0,T]}, X_2^{[0,T]}, Q_1, Q_2, \boldsymbol{Q}_3, \pi, \psi)\ p(X_1^{[0,T]} \mid Q_1)\ p(X_2^{[0,T]} \mid Q_2) \\
&= p(X_3^{[0,T]} \mid X_1^{[0,T]}, X_2^{[0,T]}, \boldsymbol{Q}_3, \pi, \psi)\ p(X_1^{[0,T]} \mid Q_1)\ p(X_2^{[0,T]} \mid Q_2) \\
&= p(X_3^{[0,T]} \mid Q_3^{[0,T]})\ p(X_1^{[0,T]} \mid Q_1)\ p(X_2^{[0,T]} \mid Q_2)
\end{aligned} \tag{3.16}$$

As mentioned before, it is plausible to marginalize out the parameters Q_1 and Q_2, for a more realistic model and inference. Noting that in case the belief state is updated using by filtering CTMPs (See Section 3.1.2.2), $Q_3^{[0,T]}$ becomes a deterministic function of all the parameters including Q_1 and Q_2, the marginalization cannot be carried out analytically on Equation 3.16. On the other hand, marginal particle filtering removes this dependency on Q_1 and Q_2 by using marginalized counterpart of CTMPs (See Section 3.1.2.2), leaving it straightforward to marginalize out the parameters on Equation 3.16.

Marginalizing the likelihood over Q_1 and Q_2:

$$\begin{aligned}
p(S^{[0,T]} \mid \pi, \psi) &= \int\int p(S^{[0,T]} \mid \theta)\ p(Q_1)\ p(Q_2)\ dQ_1 dQ_2 \\
&= \int\int p(X_3^{[0,T]} \mid Q_3^{[0,T]})\ p(X_1^{[0,T]} \mid Q_1)\ p(X_2^{[0,T]} \mid Q_2)\ p(Q_1)\ p(Q_2)\ dQ_1 dQ_2 \\
&= p(X_3^{[0,T]} \mid Q_3^{[0,T]}) \int p(X_1^{[0,T]} \mid Q_1)\ p(Q_1)\ dQ_1 \int p(X_2^{[0,T]} \mid Q_2)\ p(Q_2)\ dQ_2
\end{aligned} \tag{3.17}$$

Marginalized likelihood function for binary-valued homogenous CTMP is derived in Appendix B.

Plugging Equation B.3 in Equation 3.17 for both X_1 and X_2:

$$\begin{aligned}
p(S^{[0,T]} \mid \pi, \Phi) = p(X_3^{[0,T]} \mid Q_3^{[0,T]}) &\prod_{x_1 \in 0,1} \frac{\beta_{x_1}^{\alpha_{x_1}}}{\Gamma(\alpha_{x_1})}\ (\Upsilon(x_1) + \beta_{x_1})^{r(x_1)+\alpha_{x_1}}\ \Gamma(r(x_1) + \alpha_{x_1}) \\
&\prod_{x_2 \in 0,1} \frac{\beta_{x_2}^{\alpha_{x_2}}}{\Gamma(\alpha_{x_2})}\ (\Upsilon(x_2) + \beta_{x_2})^{r(x_2)+\alpha_{x_2}}\ \Gamma(r(x_2) + \alpha_{x_2})
\end{aligned} \tag{3.18}$$

3.2.2. Inference under Noisy Observation Model

In order to assess the robustness of the inference, we added an error probability p_e to the observation model. As explained in Section 3.1.2.1, the observation model $\psi(x_P)$ is assumed to be deterministic. This corresponds to a unique translation, denoted here by y_{true}, of each parent state x_P. In the case of noisy observation model, the resulting observation might differ from the correct translation y_{true} with probability p_e, and the erroneous translation is drawn uniformly from the remaining observation space $y_i \in \mathcal{Y} \setminus \{y_{\text{true}}\}$. This noisy observation model, denoted by ψ^{p_e}, can be considered as a noisy communication channel with error probability p_e.

$$p(y_{\text{true}} \mid x_P) = 1 - p_e \ \wedge \ p(y_j \mid x_P) = \frac{p_e}{|\mathcal{Y}| - 1} \quad \forall y_j \neq y_{\text{true}} \tag{3.19}$$

For simulation, the noisy observation model is assumed to be available to the agent, which mainly affects the agent's belief state. For exact belief state update, the noisy observation model is employed in Equation 3.11. In the case of particle filtering, the weights are assigned to the particle using the noisy model (see Algorithm 3, line 4). It is noteworthy that by doing so, the degeneration of particle filter as described in Section 3.1.2.2 is prevented.

3.2.3. Performance Metrics

As a measure of the performance of the classifier, we utilised area under the Reciever-Operator-Characteristic curve (AUROC) and Precision-Recall curve (AUPR). Since our setting presents a multi-class classification problem, the performance metrics AUROC and AUPR are evaluated as one-vs-rest.

Consider a binary classification problem with class labels 0 and 1. True positives (TP) are the samples predicted as 1 correctly, and false positives (FP) are the samples predicted as 1 while the true label was 0. True negatives (TN) are the samples predicted as 0 correctly, and false negatives (FN) are the samples with true label 1, but predicted 0. True positive rate (TPR), also called *recall* (R) or *sensitivity*, is the ratio of TPs over the number of samples which are labelled as 1. False positive rate (FPR) is the ratio of FPs over the number of samples labelled as 0. The precision (P) is the ratio of TPs over the number of all the samples predicted as 1.

$$\mathrm{TPR} = \mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3.20}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}} \tag{3.21}$$

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{3.22}$$

Receiver Operating Characteristics (ROC) curve illustrates the tradeoff between true positives and false positives, over different values of threshold for classification [13]. The area under ROC curve is a performance metric that shows how well the classifier can distinguish the classes. Higher AUROC metric indicates better performance, taking values in the interval [0,1].

Precision-Recall (PR) curve illustrates the relation between precision and recall, providing a metric to quantify how many of the predictions were correct [3].

3.3. Data Generation

The dataset contains a number of trajectories drawn from CTBN S. Following the notation in Chapter 2, K trajectories in time interval $[0, T]$ are denoted by $\xi_T^K = \left\{ S^{[0,T],1}, S^{[0,T],2}, ..., S^{[0,T],K} \right\}$, where $S^{[0,T],\kappa} = \left\{ X_1^{[0,T],\kappa}, X_2^{[0,T],\kappa}, X_3^{[0,T],\kappa} \right\}$ denotes a single trajectory for all nodes. Every trajectory comprises of state transitions in the interval, and the times of these transitions.

3.3.1. Sampling Algorithm

In order to sample trajectories from CTBN, two sampling algortihms introduced in Section 2.4 are combined. Gillespie algorithm is used to sample from the parent nodes, X_1 and X_2, while thinning algorithm is applied to overcome the challenges that come with conditional intensity matrix of the agent, X_3. It should be noted that Algorithm 1 is applicable to any nodes in a CTBN, both homogenous and conditional MPs. However, since in this setting, the intensity matrix is conditioned on the belief state and the policy, instead of directly on the parent states, a more general algorithm suitable for inhomogenous MPs, thinning algorithm, is preferred. Algorithm 4 describes the procedure to draw samples using marginal particle filtering.

Algorithm 4: Sampling trajectories with marginal particle filtering

Input : Gamma-prior parameters on parents' transition intensity matrices $\alpha^1, \beta^1, \alpha^2, \beta^2$
Set of agent's transition intensity matrices $\boldsymbol{Q}_3$
T_{max} to terminate simulation

Output : Sample trajectory of the network

Initialize: Sample Q_1 and Q_2 from their priors
Initialize nodes uniformly $X_n(0) = x_n \in \chi_n$
Initialize particles uniformly $k^i(0) = x_p \in \chi_P$
$t = 0$

while $t < T_{max}$ **do**
  Draw next transition for X_1 and X_2 (τ_{parent}, x_1 and x_2 using Algorithm 1)
  $t_{parent} \leftarrow t + \tau_{parent}$ `// transition time for parents`
  $y_{t_{parent}} \sim \psi(x_1, x_2)$ `// new observation at` t_{parent}
  Update particle filter and obtain $\mathbf{b}^{[t, t_{parent}]}$
  $a^{[t, t_{parent}]} \leftarrow \pi(\mathbf{b}^{[t, t_{parent}]})$
  $Q_3^{[t, t_{parent}]} \leftarrow Q_{3|a^{[t, t_{parent}]}}$
  $t_{agent} \leftarrow t$
  while $t_{agent} < t_{parent}$ **do**
    the upper bound for intensity, q_3^* [1]
    transition time τ_{agent} drawn by $u \sim U(0,1)$ and $\tau_{agent} = \frac{-ln(u)}{q_3^*}$
    $t_{agent} \leftarrow t_{agent} + \tau_{agent}$
    draw $s \sim U(0,1)$, accept transition if $s \leq \frac{q_3(t_{agent})}{q_3^*}$
  end while
  $t \leftarrow t_{parent}$
end while

[1] q is the transition intensity associated with the current state of the agent.

4. Results

The experimental results are presented in this chapter. First, the parameters for the variables introduced in Chapter 3 are given. Then a sample of simulated trajectories are shown as an example. Finally, the inference results are presented.

4.1. Configurations

The configurations given below are used for the results presented in the following sections, if not specified otherwise.

- Gamma priors for parent dynamics such that $Q_n \sim \mathrm{Gam}(\boldsymbol{\alpha}^n, \boldsymbol{\beta}^n)$ for $n \in \{1, 2\}$, and $\boldsymbol{\alpha}^n = [\alpha_0^n, \alpha_1^n]$ and $\boldsymbol{\beta}^n = [\beta_0^n, \beta_1^n]$

$$\boldsymbol{\alpha}^1 = [5, 10] \quad \boldsymbol{\beta}^1 = [5, 20] \tag{4.1}$$

$$\boldsymbol{\alpha}^2 = [10, 10] \quad \boldsymbol{\beta}^2 = [10, 5] \tag{4.2}$$

- Transition intensity matrices of X_1 and X_2 sampled from priors given above

$$Q_1 = \begin{bmatrix} -1.117 & 1.117 \\ 0.836 & -0.836 \end{bmatrix} \tag{4.3}$$

$$Q_2 = \begin{bmatrix} -1.1 & 1.1 \\ 2.445 & -2.445 \end{bmatrix} \tag{4.4}$$

- Length of the trajectories $T = 5$s

- State space of the parents $\mathcal{X}_P = \mathcal{X}_1 \times \mathcal{X}_2 = \{(x_1, x_2)\}_{x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2} = \{(0,0), (0,1), (1,0), (1,1)\}$

- Observation space $\mathcal{Y} = \{0, 1, 2\}$

- Action space $A = \{a_0, a_1\} = \{0, 1\}$

- The set of transition intensity matrices of X_3

$$\boldsymbol{Q}_3 = \left\{Q_{3|a_0}, Q_{3|a_1}\right\} = \left\{\begin{bmatrix} -0.5 & 0.5 \\ 2 & -2 \end{bmatrix}, \begin{bmatrix} -3 & 3 \\ 0.02 & -0.02 \end{bmatrix}\right\} \tag{4.5}$$

- Number of particles $M = 200$

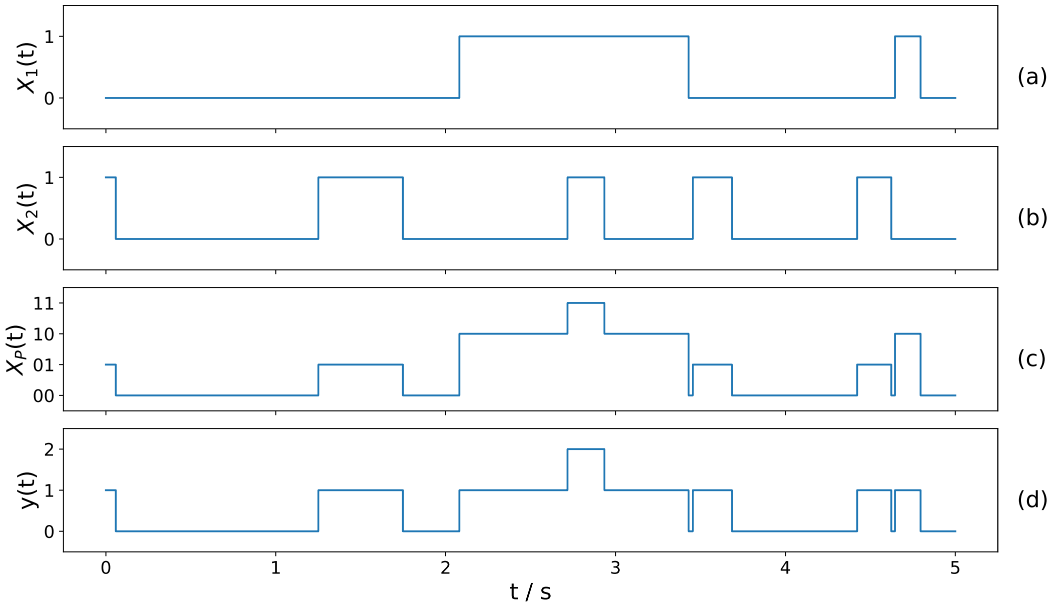

Figure 4.1.: A sample of parent trajectories and observation. (a)-(b) A sample trajectory of parent nodes X_1 and X_2 of length $T = 5\text{s}$, (c) The trajectory of the joint parent process X_P, (d) The observation trajectory resulting from X_P given in (c) and ψ_{true} given in Section 4.1.

- Weights of the policy, introduced in Equation 3.13, $w = [0.02, 0.833, 0.778, 0.87]$
- Observation model $\psi_{\text{true}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

4.2. Simulation

The synthetic dataset is generated utilising Algorithm 4. K trajectories in time interval $[0, T]$ are denoted by $\xi_T^K = \left\{S^{[0,T],1}, S^{[0,T],2}, ..., S^{[0,T],K}\right\}$, where $S^{[0,T],\kappa} = \left\{X_1^{[0,T],\kappa}, X_2^{[0,T],\kappa}, X_3^{[0,T],\kappa}\right\}$ denotes a single trajectory for all nodes. It is noteworthy that the initial states are drawn from disrete uniform distribution.

$$X_n(0) \sim \mathcal{U}\{0, 1\} \text{ for } n \in \{1, 2, 3\} \tag{4.6}$$

Figure 4.1(a)-(b) shows an example of parent trajectories. In Figure 4.1(c), the resulting trajectory of the joint parent process X_P is illustrated. As mentioned in Section 3.1.1, this joint process over the parent nodes provides a compact representation. In Figure 4.1(c), the states of X_P taking values in $\mathcal{X}_P = \mathcal{X}_1 \times \mathcal{X}_2$ is preferred to be represented as a combination of the parent states for readibility, so that $\mathcal{X}_P = \{00, 01, 10, 11\}$, where $x_P \in \mathcal{X}_P$ corresponds to

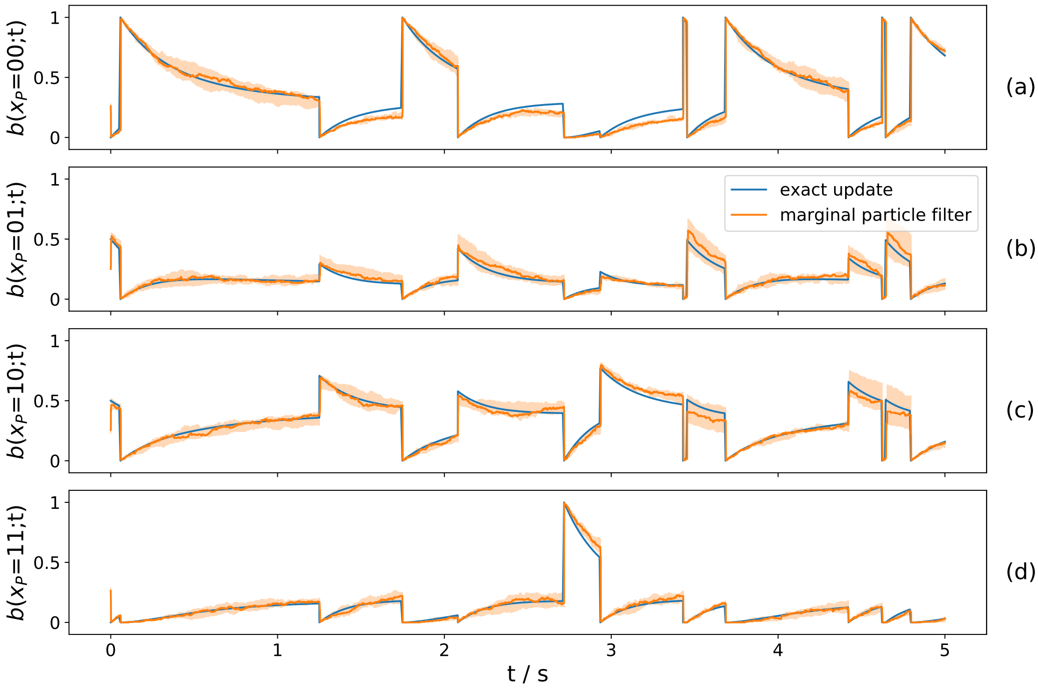

Figure 4.2.: Belief state trajectories corresponding to the observations given in Figure 4.1(d), comparing exact update method and marginal particle filtering.

$x_1x_2,\ x_1 \in \mathcal{X}_1,\ x_2 \in \mathcal{X}_2$. Figure 4.1(d) shows the observation trajectory resulting from $X_P(t)$ and the observation model ψ_{true} given in Section 4.1.

Figure 4.2 illustrates the belief state trajectory given the observations in Figure 4.1(d). For the reference, the belief state update using marginal particle filter and exact update are given together. Exact update of belief state is obtained as described in Section 2.3.1, while the marginal particle filter update is the result of sequential implementation of Algorithm 3. For marginal particle filter, we plot the median as a line over 10 runs, and 25-75th percentile as the shaded area. As can be seen from the figures, the exact update is well approximated by the marginal particle filter. Finally, the resulting Q_3 and the trajectory of the agent are given in Figure 4.3. The Q_3 trajectory shown in Figure 4.3(b) is derived from the belief state update by marginal particle filter which is given in Figure 4.3(a) using Equation 3.13 and Equation 3.15.

As mentioned in Section 3.1.2.2, the degeneration of the marginal particle filter in case of unlikely changes in observation, i.e. rapid transitions of parent nodes, is handled by assigning uniform weights to the particles. It effectively corresponds to ignoring these changes in the observation, which may cause divergence from the exact update. However, it is recovered with the next observation. Figure 4.4 provides an example of the situation. The rapid change in question is highlighted in Figure 4.4(a). The particles fail to simulate this observation. The divergence can be observed in Figure 4.4(d)-(e) clearly.

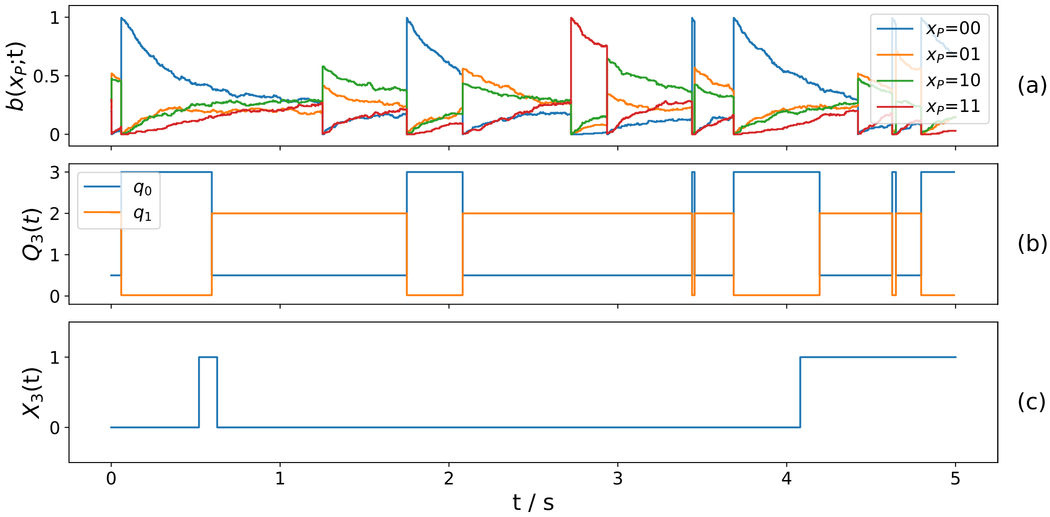

Figure 4.3.: Belief state updated by marginal particle filter and the resulting Q_3 and X_3 trajectories.

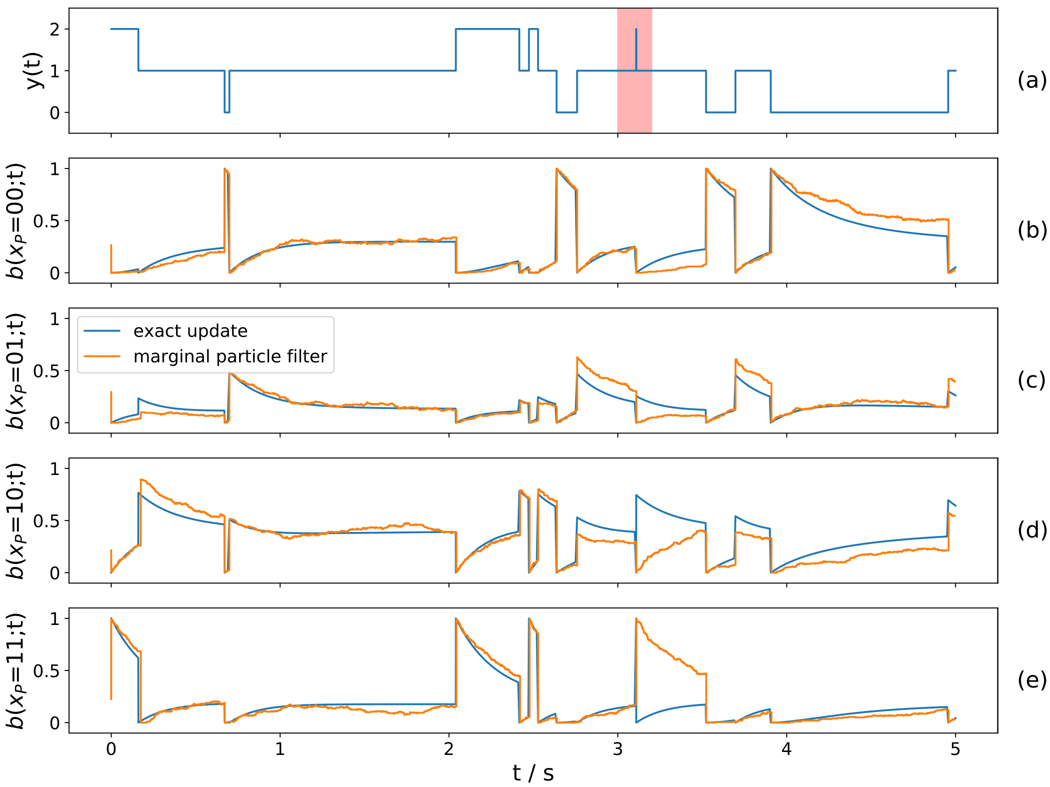

Figure 4.4.: A sample with degenerate marginal particle filter. The unlikely observation which has caused the degeneration is highlighted in (a). It can be seen in (d) and (e) that this observation causes marginal particle filter approximation to diverge from exact update results but it is recovered with the next observation.

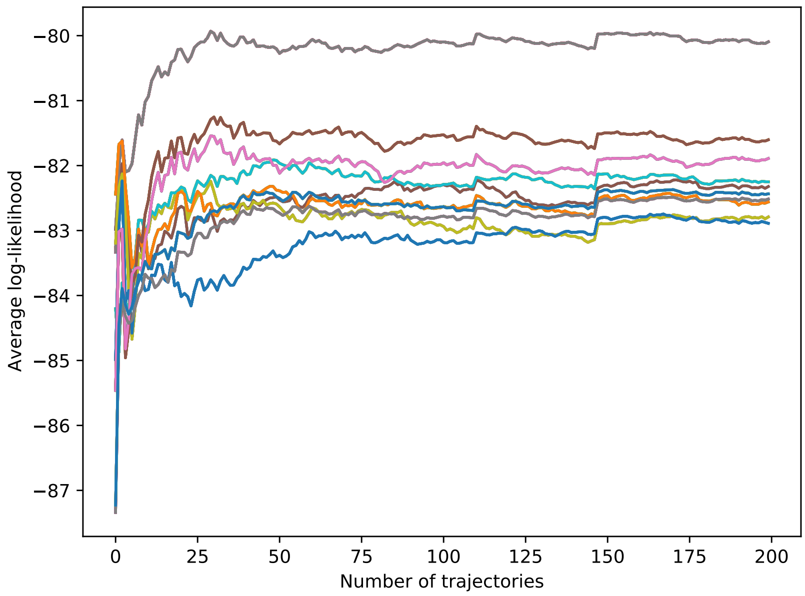

Figure 4.5.: Average log-likelihood $\log p(S^{[0,T]} \mid \theta)$ over samples generated using exact belief state update, depicting the equivalence classes in the set of observation models.

4.3. Inference of Observation Model

4.3.1. Equivalence Classes

As mentioned in Section 3.2, the deterministic nature of the observation model results in a number of possible observation models. The setting described in Chapter 3, with configurations given in Section 4.1, leads to 81 observation models. However, with this experimental setup and the methods, it is only possible to distinguish these observation models into 10 different classes. Due to this equivalence, the inference problem is considered only for 10 observation models, each one representing one class. The origins of this phenomena are discussed in detail in Appendix C, together with the observation models considered in the inference problem. The set of observation model that can be classified is denoted as $\boldsymbol{\psi}$ in the following.

Figure 4.5 illustrates the equivalence of observation models clearly. The plot depicts the results of an experiment with 200 samples, ξ_T^{200}, generated using the observation model ψ_{true} given in Section 4.1, and the average log-likelihood of samples computed for all possible observation models. Here, the belief state is updated using exact method as described in Section 3.1.2.2, in order to depict the exact equivalence within one class. As can be seen, the results show the separation of the set of observation models into 10 distinct classes. The legend is removed to avoid clutter. The jump around 110 and 150 trajectories can be explained by an encounter with a sample of highly likely parent trajectories. Such encounters have the same affect on the likelihood of all observations, as can be seen from Equation 3.18.

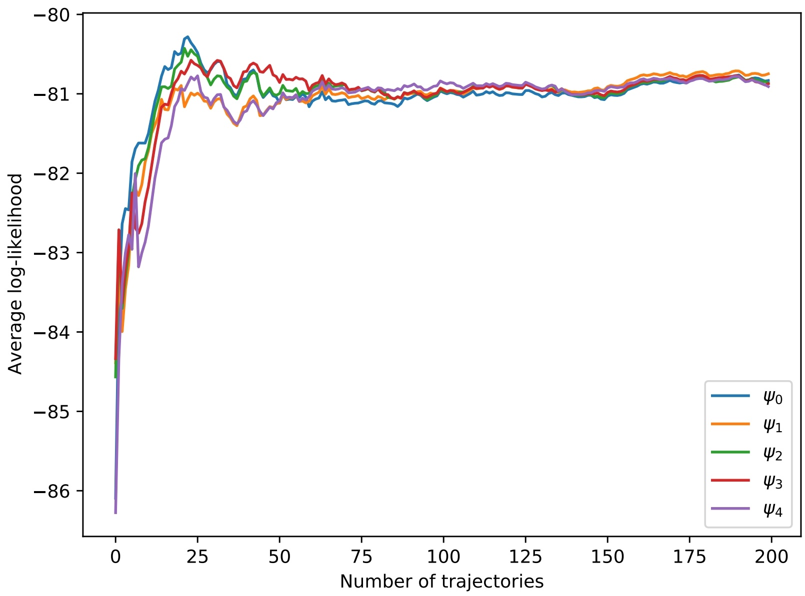

Figure 4.6.: Average log-likelihood $\log p(S^{[0,T]} \mid \theta)$ over samples generated using marginal particle filtering, where ψ_1 and ψ_2 belongs in the same class.

In order to show the validity of the equivalence in the case of marginal particle filter, the average log-likelihoods of 200 samples given two observation models in the same class are illustrated in Figure 4.6. The samples are generated with $\psi_{\text{true}} = \psi_0$, and the rest of the observation models here fall in the same equivalence class as ψ_0. As can be seen from the graph, the observation models lead to such similar results that they are assumed to be identical.

4.3.2. Learning Observation Model

Figure 4.7 illustrates the average log-likelihood over 100 samples given the observation models. The samples are generated with $\psi_{true} = \psi_0$ given in Section 4.1, and exact update method is utilised for belief state update. As can be seen, the curves converge quickly and the true model is well separated from others. Consequently, the maximum log-likelihood estimation, as in Equation 4.7, leads to the correct result. Here, in order to avoid numerical unstability, the log-likelihood is preferred for the calculations instead of likelihood.

$$\hat{\psi} = \arg\max_{\psi} \left[\log p(S^{[0,T]} \mid \theta)\right] \tag{4.7}$$

A similar experiment is documented in Figure 4.8, where exact update method is replaced by marginal particle filtering. ψ_0 denotes the observation model that has generated the dataset, and it is correctly estimated as the true model. The jump around 50 trajectories can be explained by an encounter with a sample of highly likely parent trajectories.

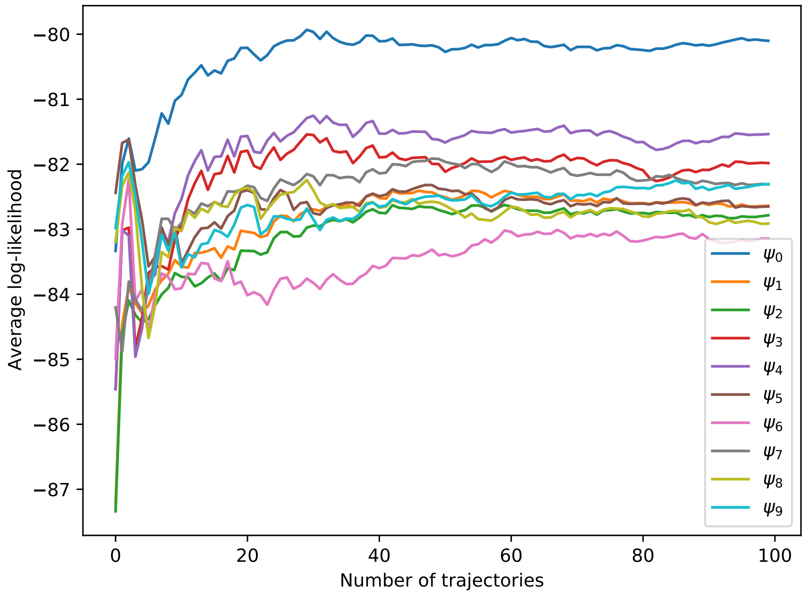

Figure 4.7.: Average log-likelihood $\log p(S^{[0,T]} \mid \theta)$ with $\psi_i \in \boldsymbol{\psi}$ over samples generated using exact belief state update

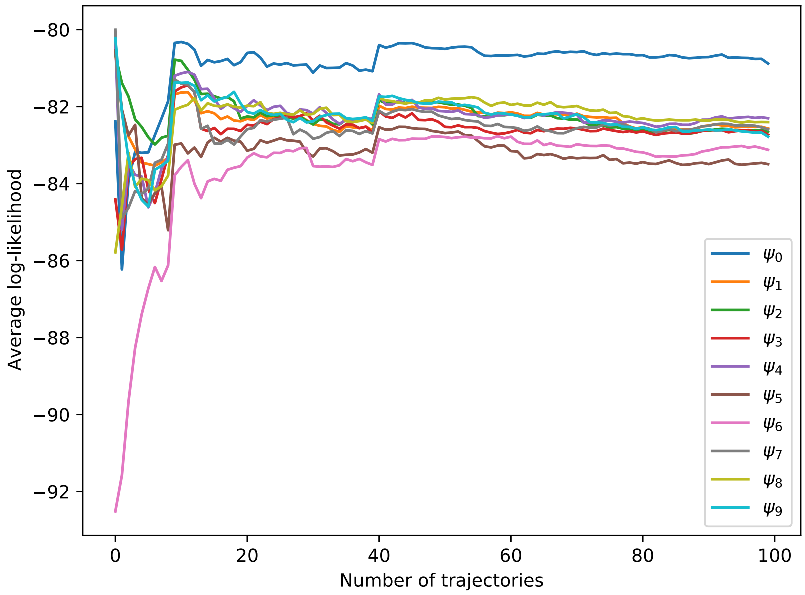

Figure 4.8.: Average log-likelihood $\log p(S^{[0,T]} \mid \theta)$ with $\psi_i \in \boldsymbol{\psi}$ over samples generated using marginal particle filtering

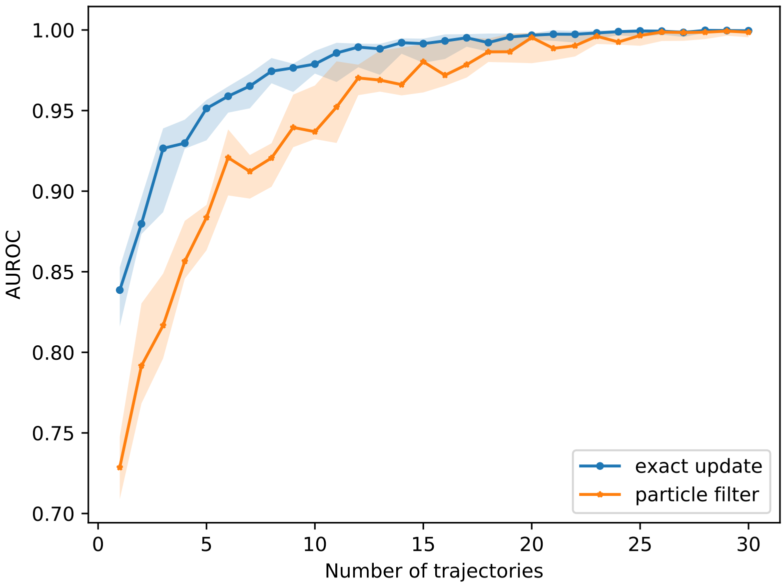

Figure 4.9.: AUROC results over an increasing number of samples for ψ_0-vs-rest. We plot the median with a line and the 25-75th percentile with the shaded area over 10 runs.

We approach the inference problem as classification between the observation models $\psi \in \boldsymbol{\psi}$ given in Equation (C.7), and consider the estimated likelihood values of each sample given an observation model as the score of the sample belonging to the corresponding class. As a measure of the performance of the classifier, we utilise area under the Reciever-Operator-Characteristic curve (AUROC) and Precision-Recall curve (AUPR). Since this setting represents a multi-class classification problem, the performance metrics AUROC and AUPR are calculated as one-vs-rest.

We have provided the classifier with increasing number of samples for inference. This is achieved through bootstrapping a given number of trajectories, and using the mean likelihood over the bootstrap batch as a new sample. The following AUROC plots shows the results over 50 trajectories generated using each obervation model as the true model. According to this, in our dataset, we have 500 trajectories, 50 from each class labeled through a vector with 10 entries, having a single element equal to 1 for the true observation model and 0 for the rest. When number of trajectories is 1, each sample in the dataset is considered individually. When number of trajectories is 2, within each class, 50 sample batches of size 2 are bootstrapped such that none of the batches consist of the same samples. By doing so, we keep the sample size at 50 per class, regardless of the batch size.

Figure 4.9 shows the AUROC results over 10 runs, comparing the state estimator using exact update and marginal particle filtering. We plot the median as a line and 25-75th percentile as the shaded area. The AUPR results are given in appendix Figure D.1 in the same manner. As expected given the unbiased classifier, both metrics approach to 1 as the number of samples

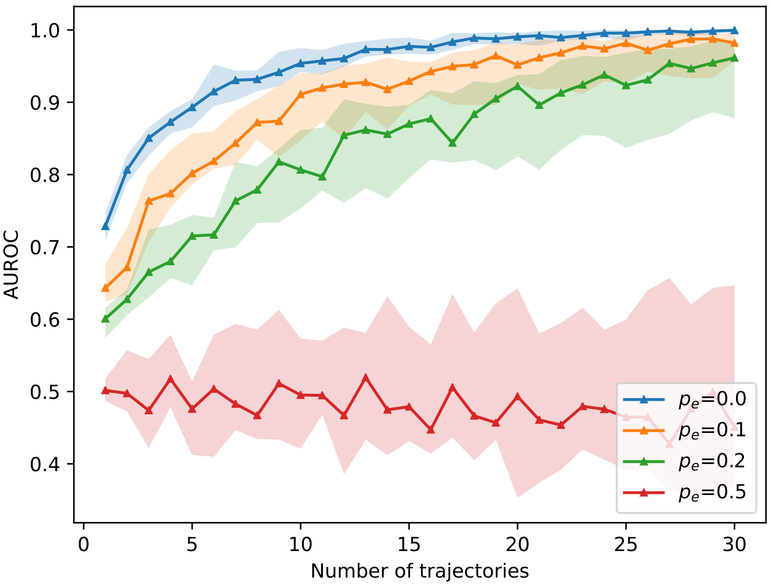

Figure 4.10.: AUROC results over an increasing number of samples for ψ_0-vs-rest. We plot the median with a line and the 25-75th percentile with the shaded area over 10 runs. The performance deteriorates as the noise increases, however, with the increasing number of trajectories the metric converges to 1, showing the robustness.

increases. Due to the fact that less information is available to the marginal particle filter as the state estimator, the results obtained with this method show slightly lower performance.

4.3.3. Inference under Noisy Observation Model

We have experimented with some added noise to the observation model, to test the robustness of the inference. How this noise was introduced to the observation model and how it affects the resulting observation is discussed in Section 3.2.2. The noise is quantified with an error probability p_e of the observation model leading to a wrong observation. The noisy observation model can be interpreted as a noisy communication channel with an error probability of p_e.

AUROC results for different levels of noise are given in Figure 4.10. We plot the median as a line and 25-75th percentiles are shown as the shaded area around. The performance decreases as the noise introduced to the true observation model increases. These results confirm the expectations, as the noise leads to less reliable observations for the agent. On the other hand, with the increasing number of trajectories the metric converges to 1, showing robustness. The error probability of 0.5 shows the breaking point, at which the classes are no longer separable for the classifier.

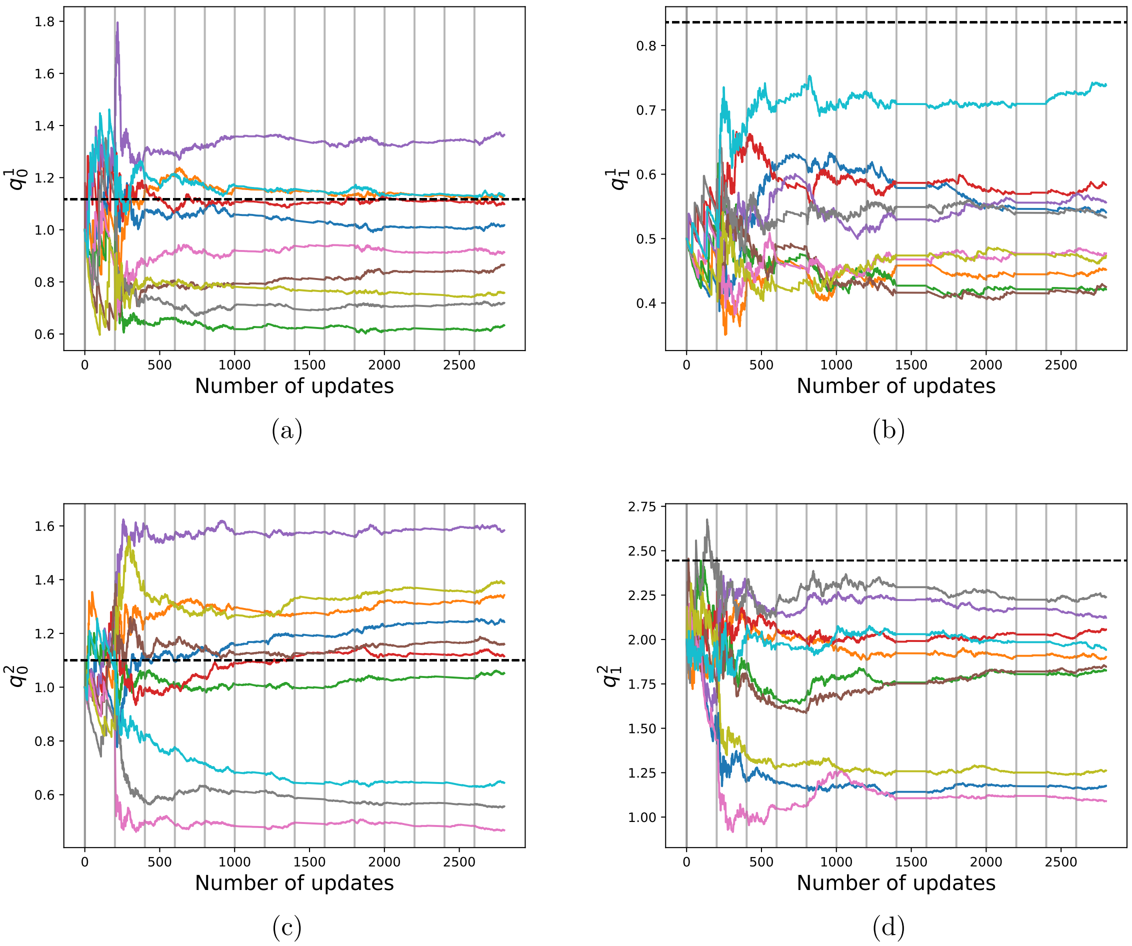

Figure 4.11.: Estimated transition intensities of Q_1 and Q_2 by marginal particle filter. Each colored line illustrates one estimation from 10 particle filters shown in Figure 4.2. The notation on y-axis label follows the notation in Equation 3.1. Black dashed line marks the true values of the transition intensity. The gray lines show the observations. Between each observation, there are 200 updates to the estimation, each coming from one particle. The legend is removed to avoid clutter but each color corresponds to the same marginal particle filter in each plot.

4.3.4. Estimated Intensity Matrices by Marginal Particle Filter

In marginal particle filtering, the belief state is formed over particles drawn from marginalized processes of parents as in Equation 3.12. In order to achieve this, the intensity matrices are replaced with the expectation given in Equation 2.39. With every new observation, the particles are propagated through the marginal process and the summary statistics are updated after each particle. Gamma-priors of the parent transition intensities are given in Section 4.1.

In Section 4.2, we have presented an example of sampled trajectories. Figure 4.1 illustrates the parent trajectories, and Figure 4.2 shows the belief state updated using exact method, and marginal particle filtering. For the latter, we plotted the statistics over 10 filters. Figure 4.11 illustrates the estimated transition intensities of those 10 particle filters. The notation on y-axis label follows the notation in Equation 3.1. Each colored line illustrates one estimation from 10 particle filters shown in Figure 4.2. Black dashed line marks the true values of the transition intensity. The gray lines show the observations. Between each observation, there are 200 updates to the estimation, each received from one particle.

As can be seen from the figures, the estimations converge, although the final estimated value might be different than the true values. This can be explained by the fact that at the early stages, an unlikely transition sampled from marginal process causes a significant change in the estimation, which in return affects the distribution from which the next transitions are drawn. Such significant ripples can be seen at the early updates, i.e. updates 0 to 500 in the course of first 3 observations, which play a major role on the value the estimation converges to.

5. Conclusion

Motivated by the examples of multi-agent systems in nature, we modelled a communication between two parent nodes and an agent node combining CTBN and POMDP frameworks. While the parent nodes emit messages containing information about their states, the agent observes a translation of these messages from which it needs to form its belief state and make decisions. The nodes evolve continuously in time as components of a CTBN, modelled as in Section 3.1.1. Given that the messages of the parent nodes are unavailable to the agent, the interaction between the parent nodes and the agent node is modelled as POMDP, as described in Section 3.1.2. We infer the observation model from simulated trajectories of nodes.

The belief state is updated utilising two methods. The first one is the exact update method, discussed in Section 3.1.2.2, and assumes that the transition intensities of the parents Q_1 and Q_2 are available for both the agent and the classifier. However, since this would not present a realistic system, particle filtering with marginalized CTBN is introduced as the state estimator. Given Gamma-priors of Q_1 and Q_2, the exact update method is well approximated by the marginal particle filter.

We consider this problem as a classification between observation models and analyze the performance of the classifier in terms of the metrics AUROC and AUPR. The results are given in Figure 4.9 and Figure D.1, respectively. Using the exact method to update the belief state, excellent performance is achieved for the classification task regarding AUROC metric. Since less information is available to the marginal particle filter, it yields a slightly lower performance, compared to the exact update method. Nevertheless, in both methods, as the number of samples increases, the metric approaches to 1, which is expected in the case of the unbiased classifier.

The results of the experiments with different levels of noise added to the observation model are given in Figure 4.10. We assume that the noise level is available to the agent, and the likelihoods of noise-free observation models are considered for classification. Thus, the noisy model parameters are not estimated. The performance decreases as the noise introduced to the true observation model increases. These results confirm the expectations, as the noise leads to less reliable observations for the agent. On the other hand, with the increasing number of trajectories the metric converges to 1, showing robustness.

An important limitation to the inference task is the equivalence classes as introduced in Section 4.3.1. The set of observation models can be divided into 10 equivalence classes such that the likelihoods of a sample trajectory $S^{[0,T]}$ given any observation model within one class are equal. This situation limits the classifiers ability to determine the true model. Due to this limitation, the set of observation models is reduced to 10 observation models, each one representing an equivalence class. These observation models are given in Equation (C.7). Consequently, the result which states that the true observation model is ψ_i is equivalent to the one which states that the true observation model belongs to i$^{\text{th}}$ equivalence class. As shown with examples in Appendix C, there are two reasons behind this equivalence. The first reason can be specified as the equivalent effect of observation models on the belief state, which is inherent to the observation model structure. The second reason is that the different belief states might lead to the same behaviour. This case can intuitively be explained by the fact that the agent may not need to use all the information it has for decision making. This information loss limits the ability to infer what the agent has observed from its actions.

5.1. Outlook

A major step in the future is to eliminate the equivalence classes to be able to classify every observation model. This problem, to a certain extent, can be mitigated by joint inference of observation model and policy. The joint inference could be performed as a joint classification problem, where the combinations of discrete values of these parameters are treated as classes. This is only feasible by defining appropriate constraints on the policy such that, as for the case of observation models in this work, the policy space is countable. Another approach is to use function approximation to learn the observation model and the policy jointly.

Another exciting direction is to employ our method in different environments to get insights into the interactions of agents and environments. Foerster *et al.* [7] study a problem where the agents must learn to a communication protocol for information sharing to coordinate over a task. Our method can be utilised to infer and analyse the communication protocols that lead to the success or failure of the agents. For example, in the multi-step MNIST game, where the agents should agree on an encoding of digits, the encoding on which they have reached a consensus can be inferred. However, the problem needs simplifications to be feasible with our assumptions, i.e. only one acting agent.

Moreover, it would be interesting to apply our model and solution approach to a more complex environment to evaluate the performance further. For example, one could experiment with non-binary messages, or more than two parent nodes.

In this thesis, we assume that the messages are relevant for a single node, and its actions affect merely its own dynamics. An extension to this model is to consider bidirectional communication between agents, as depicted in Figure 5.1. In such communication, both nodes send messages, which are translated by the same observation model, and their behaviour depends on their observations and the belief states they keep.

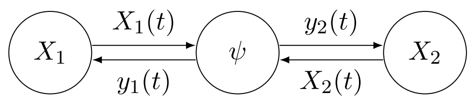

Figure 5.1.: Multi-agent communication model.

Appendices

A. Amalgamation Operation

A CTBN with multiple variables can be represented with a single intensity matrix. This is done by amalgamation operation. Amalgamation defines a combining operation over multiple intensity matrices and produces a single matrix for the entire system [18].

Amalgamation of Independent Processes

Consider a CTBN with graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ over two variables such that $\mathcal{V} = \{X_1, X_2\}$. Assume variables X_1 and X_2 are parent nodes, with intensity matrices Q_1 and Q_2, i.e. $\mathcal{E} = \emptyset$ and $Par_{\mathcal{G}}(X_1) = Par_{\mathcal{G}}(X_1) = \emptyset$. This CTBN can be identified as a subsytem of the CTBN model described in Section 3.1.1.
Analogous to Equation 2.5, Markov transition function for the joint process can be derived as

$$\begin{aligned}\Pr(X_P(t+h) = x'_p \mid X_P(t) = x_p) &= \Pr(X_1(t+h) = x'_1, X_2(t+h) = x_2 \mid X_1(t) = x_1, X_2(t) = x_2) \\ &= \Pr(X_1(t+h) = x'_1 \mid X_1(t) = x_1, X_2(t) = x_2) \\ &\quad\ \Pr(X_2(t+h) = x_2 \mid X_1(t) = x_1, X_2(t+h) = x_2) \\ &= (\delta_{x'_1,x_1} + hq^1_{x_1,x'_1} + o(h))(1 + hq^2_{x_2,x_2} + o(h)) \\ &= \delta_{x'_1,x_1} + hq^1_{x_1,x'_1} + h\delta_{x'_1,x_1} q^2_{x_2,x_2} + o(h) \end{aligned} \tag{A.1}$$

where $x_1, x'_1 \in \mathcal{X}_1$, $x_2, x'_2 \in \mathcal{X}_2$, $x_p = (x_1, x_2), x'_p = (x'_1, x_2) \in \mathcal{X}_P$.
Suppose the intensity matrices of X_1 and X_2 are in the form

$$Q_i = \begin{bmatrix} -q^i_0 & q^i_0 \\ q^i_1 & -q^i_1 \end{bmatrix} \quad \text{for } i \in \{1, 2\} \tag{A.2}$$

Then the intensity matrix for the joint process X_P with factorising state space $\mathcal{X}_P = \mathcal{X}_1 \times \mathcal{X}_2$ can be written as

$$Q_P = \begin{bmatrix} -q^2_0 - q^1_0 & q^2_0 & q^1_0 & 0 \\ q^2_1 & -q^2_1 - q^1_0 & 0 & q^1_0 \\ q^1_1 & 0 & -q^1_1 - q^2_0 & q^2_0 \\ 0 & q^1_1 & q^2_1 & -q^1_1 - q^2_1 \end{bmatrix} \quad \text{for } i \in \{1, 2\} \tag{A.3}$$

As it can be observed from Equation A.3, the transition intensities which correspond to state transition in both variables, i.e. anti-diagonal entries, are zero, due to one of the assumptions in CTBN framework that only one variable can transition at a time, as given in Section 2.2.

B. Marginalized Likelihood Function for Homogenous Continuous Time Markov Processes

Let X be a homogenous CTMP. For convenience, it is assumed to be binary-valued, $\mathcal{X} = \{x_0, x_1\}$. The transition intensity matrix can be written in the following form:

$$Q = \begin{bmatrix} -q_0 & q_0 \\ q_1 & -q_1 \end{bmatrix} \tag{B.1}$$

where the transition intensities q_0 and q_1 are gamma-distributed with parameters α_0, β_0 and α_1, β_1, respectively. The marginal likelihood of a sample trajectory $X^{[0,T]}$ can be written as follows:

$$\begin{aligned}
P(X^{[0,T]}) &= \int P(X^{[0,T]} \mid Q)P(Q)dQ \\
&= \int_0^\infty = \prod_{j \neq i} \exp(-q_{i,j}\Upsilon(x_i))\ q_{i,j}^{r(x_i,x_j)} \frac{\beta_{i,j}^{\alpha_{i,j}} q_{i,j}^{\alpha_{i,j}-1} \exp(-\beta_{i,j}q_{i,j})}{\Gamma(\alpha_{i,j})}\ dq_{i,j} \\
&= \prod_{i \in 0,1} \int_0^\infty q_i^{r(x_i)}\ \exp(-q_i\Upsilon(x_i))\ \frac{\beta_i^{\alpha_i}\ q_i^{\alpha_i-1}\ \exp(-\beta_i q_i)}{\Gamma(\alpha_i)}\ dq_i \\
&= \prod_{i \in 0,1} \frac{\beta_i^{\alpha_i}}{\Gamma(\alpha_i)} \int_0^\infty q_i^{r(x_i)+\alpha_i-1}\ \exp(-q_i(\Upsilon(x_i)+\beta_i))\ dq_i & \text{(B.2)} \\
&= \prod_{i \in 0,1} \frac{\beta_i^{\alpha_i}}{\Gamma(\alpha_i)} \Big(-(\Upsilon(x_i)+\beta_i)^{-r(x_i)-\alpha_i}\ \Gamma(r(x_i)+\alpha_i,\ q_i(\Upsilon(x_i)+\beta_i))\Big)\Big|_0^\infty \\
&= \prod_{i \in 0,1} \frac{\beta_i^{\alpha_i}}{\Gamma(\alpha_i)} \Big((\Upsilon(x_i)+\beta_i)^{-r(x_i)-\alpha_i}\ \Gamma(r(x_i)+\alpha_i)\Big) & \text{(B.3)}
\end{aligned}$$

In Equation B.2, the integral is solved using computer algebra system WolframAlpha as follows:

$$\int x^a\ \exp(-xb)\ dx = -b^{-a-1}\ \Gamma(a+1,\ bx) + C \tag{B.4}$$

C. Equivalence Classes of Observation Models

The equivalence classes are inherent to the problem setting and caused by two reasons. In this appendix, these reasons are explained and illustrated.

Identical Effect on Belief State

Some observation models fall into the same class as the effect they have on the belief state is identical. An example of this situation is illustrated below. In order to show the exact equivalence, the simulations employ the exact update method as described in Section 3.1.2.2. Consider the problem of calculating the likelihood of one sample $S^{[0,T]}$ given two observation models, ψ_1 and ψ_2 as in Equations (C.1)-(C.2). Given a sample of parent trajectories shown in Figure C.1, it is obvious that these two observation models lead to different observation trajectories as shown in Figure C.2(a) and Figure C.3(a). Nonetheless, using Equation 3.11, the resulting belief state is exactly the same. This leads to the exact same trajectories for Q_3 and the likelohood of the sample given these two observation models, $p(S^{[0,T]} \mid \psi_1)$ and $p(S^{[0,T]} \mid \psi_2)$ end up being exactly the same.

$$\psi_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad p(S^{[0,T]} \mid \psi_1) = -83.334 \tag{C.1}$$

$$\psi_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad p(S^{[0,T]} \mid \psi_2) = -83.334 \tag{C.2}$$

Suppose $X_P(t_0) = 00$ and the initial belief state is $b(t_0^-)$ is uniformly distributed. This joint parent state leads to $y_1(t_0) = y_0$ with observation model ψ_1, and $y_2(t_0) = y_1$ with ψ_2. It should be noted that in the observation model, each column corresponds to the probability of an observation given each state. Therefore, the first column of observation model can be denoted as $c_{y_0} = \{p(y = y_0|x_P)\}_{x_P \in \mathcal{X}_P}$. From Equation 3.11, the new belief state is obtained as the element-wise multiplication of initial belief state $b(t_0^-)$ and the column corresponding to the received observation.

$$b_1(t_0) = Z^{-1}\, c_{y_0} \circ b(t_0^-) = Z^{-1} \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \circ \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \tag{C.3}$$

$$b_2(t_0) = Z^{-1}\, c_{y_1} \circ b(t_0^-) = Z^{-1} \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \circ \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \tag{C.4}$$

where Z is the normalization factor, $b_1(t)$ and $b_2(t)$ are the belief states corresponding to ψ_1 and ψ_2, respectively. As can be seen, even though the observations are different, ψ_1 and ψ_2

lead to the same belief state update. This derivation can be done similarly for other state values $X_P(t_0)$.

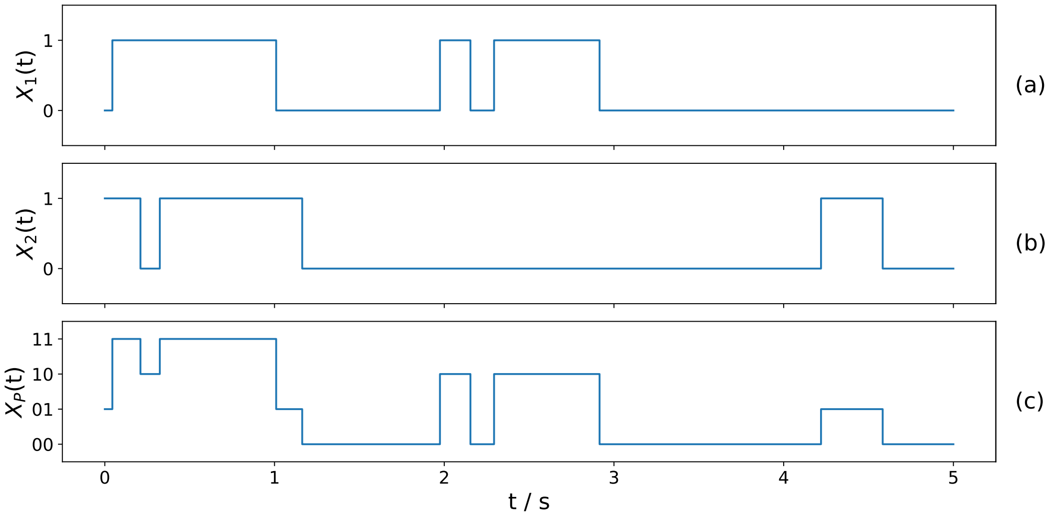

Figure C.1.: Parent trajectories for the models leading to the same belief state

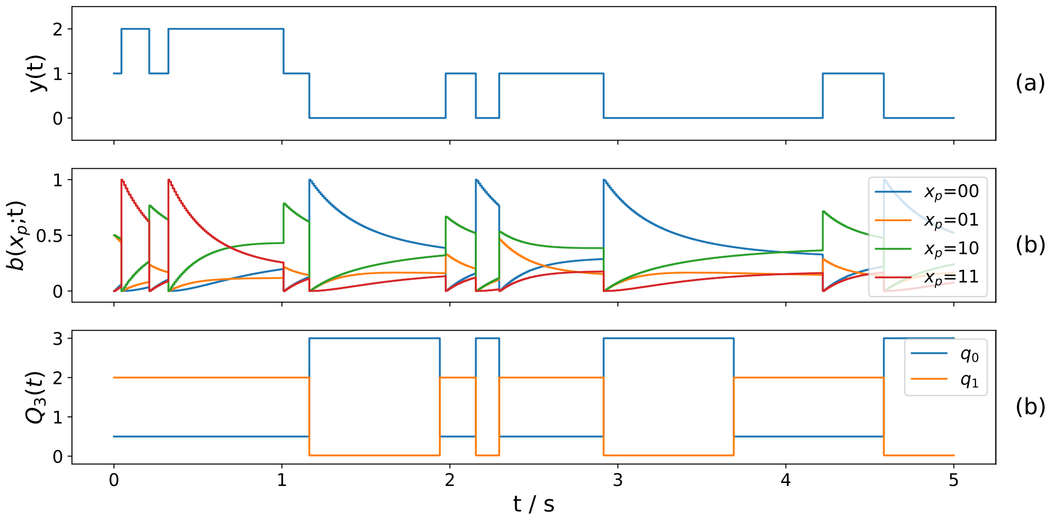

Figure C.2.: Observation, belief state and Q_3 trajectories derived by ψ_1 in Equation C.1 corresponding to parent trajectories in Figure C.1

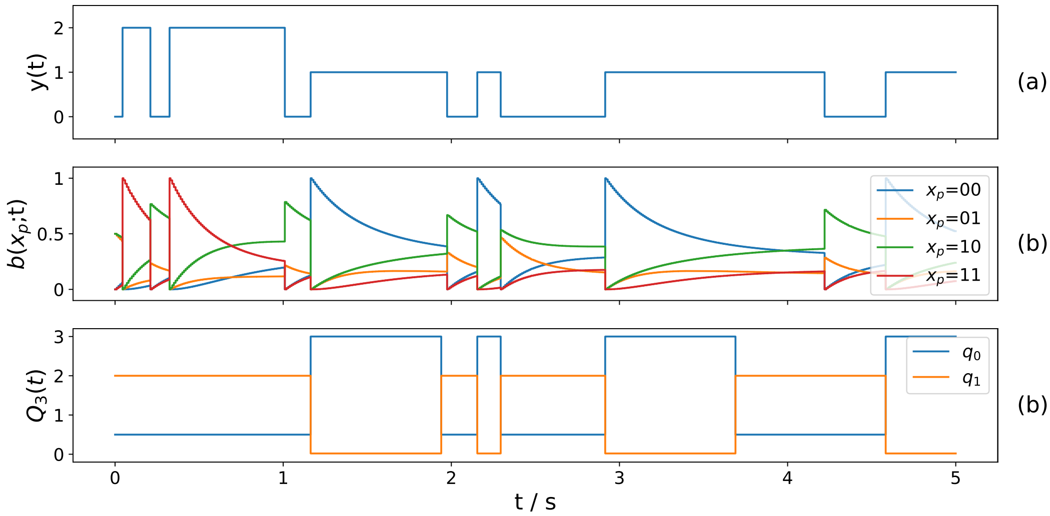

Figure C.3.: Observation, belief state and Q_3 trajectories derived by ψ_2 in Equation C.2 corresponding to parent trajectories in Figure C.1

Combination of Belief State and Policy

For some observation model, the reason of equivalence is that even though the belief state are different, the policy $\pi(b)$ leads to same trajectory for Q_3. This case is exemplified below where the simulations are employ the exact update method as described in Section 3.1.2.2. Consider the problem of calculating the likelihood of one sample $S^{[0,T]}$ given two observation models, ψ_1 and ψ_2 as in Equations (C.5)-(C.6). Given a sample of parent trajectories shown in Figure C.4, these observation models lead to different observation trajectories as shown in Figure C.5(a) and Figure C.6(a). These trajectories result in different belief state trajectories as in Figure C.5(b) and Figure C.6(b). However, the policy leads to the exact same trajectories for Q_3 and the likelihood of the sample given these two observation models, $p(S^{[0,T]} \mid \psi_1)$ and $p(S^{[0,T]} \mid \psi_2)$ end up exactly same.

$$\psi_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad p(S^{[0,T]} \mid \psi_1) = -80.648 \tag{C.5}$$

$$\psi_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad p(S^{[0,T]} \mid \psi_2) = -80.648 \tag{C.6}$$

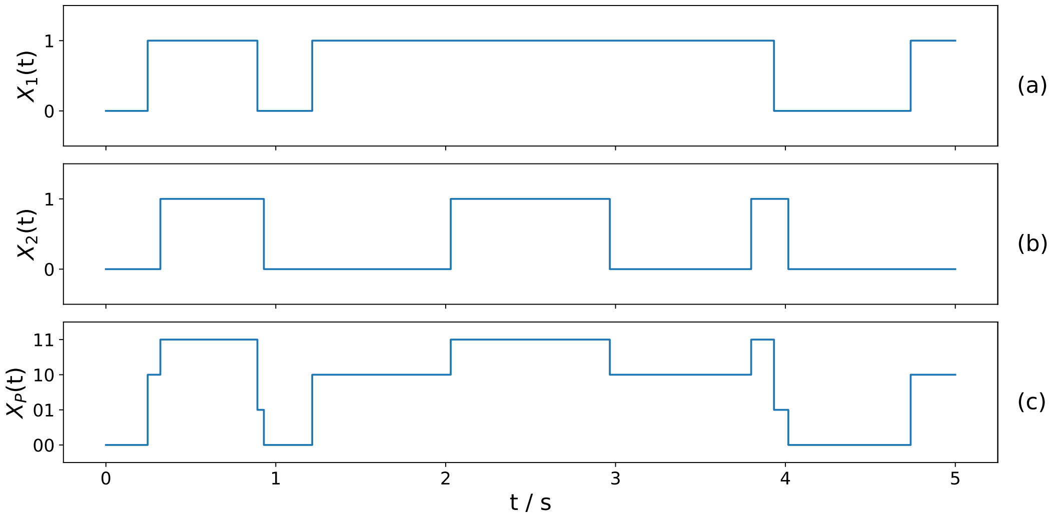

Figure C.4.: Parent trajectories for the models leading to the same behaviour

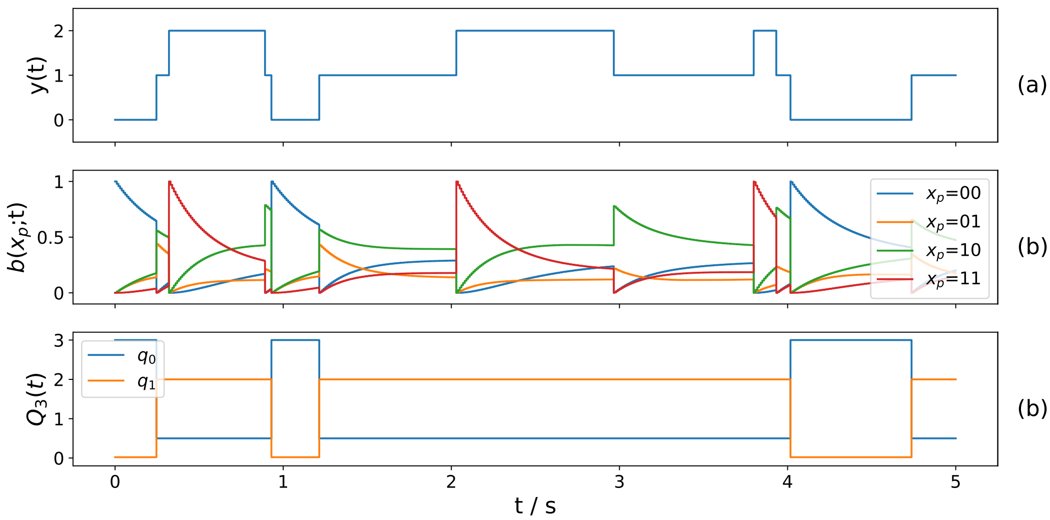

Figure C.5.: Observation, belief state and Q_3 trajectories derived by ψ_1 in Equation C.5 corresponding to parent trajectories in Figure C.4

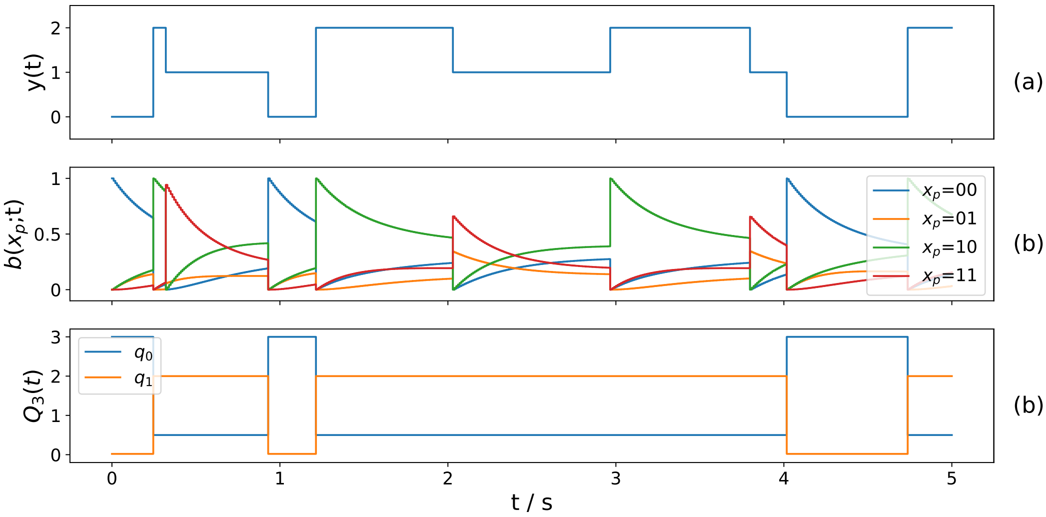

Figure C.6.: Observation, belief state and Q_3 trajectories derived by ψ_2 in Equation C.6 corresponding to parent trajectories in Figure C.4

Observation Models in Experiments

As mentioned in Section 4.3.1, the inference problem is reduced to maximum likelihood estimation between 10 classes. We have selected one observation model as representative of each class and considered for the inference problem. The representative observation models are given below. This set of observation models are referred to as $\boldsymbol{\psi}$.

$$\psi_{\text{true}} = \psi_0 = \begin{bmatrix}1&0&0\\0&1&0\\0&1&0\\0&0&1\end{bmatrix} \psi_1 = \begin{bmatrix}1&0&0\\1&0&0\\1&0&0\\0&0&1\end{bmatrix} \psi_2 = \begin{bmatrix}0&1&0\\0&1&0\\1&0&0\\0&1&0\end{bmatrix} \psi_3 = \begin{bmatrix}0&1&0\\0&1&0\\1&0&0\\1&0&0\end{bmatrix} \psi_4 = \begin{bmatrix}0&1&0\\0&1&0\\1&0&0\\0&0&1\end{bmatrix}$$
$$\psi_5 = \begin{bmatrix}0&1&0\\1&0&0\\0&1&0\\0&1&0\end{bmatrix} \psi_6 = \begin{bmatrix}0&1&0\\1&0&0\\0&1&0\\1&0&0\end{bmatrix} \psi_7 = \begin{bmatrix}0&1&0\\1&0&0\\0&1&0\\0&0&1\end{bmatrix} \psi_8 = \begin{bmatrix}0&1&0\\1&0&0\\1&0&0\\0&1&0\end{bmatrix} \psi_9 = \begin{bmatrix}0&1&0\\1&0&0\\0&0&1\\0&1&0\end{bmatrix} \tag{C.7}$$

D. Additional Results

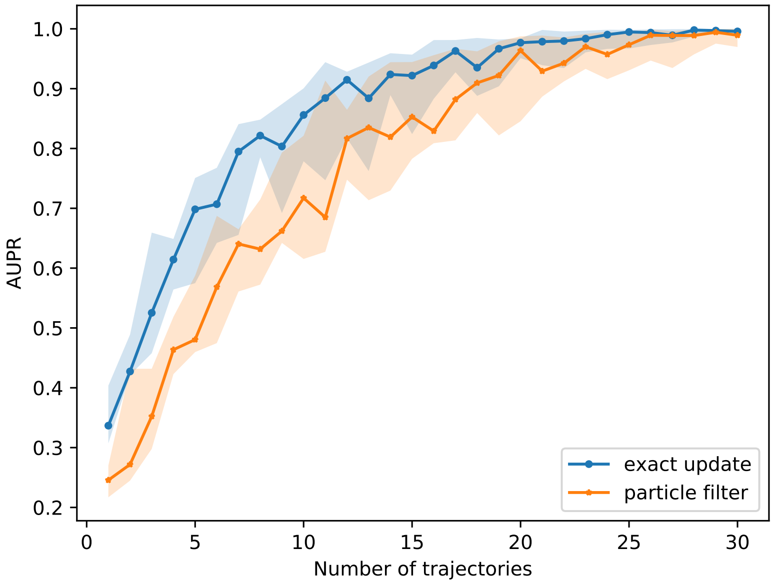

Figure D.1.: AUPR results over increasing number of samples for ψ_0-vs-rest. We plot the median with a line and the 25-75th percentile with the shaded area over 10 runs.

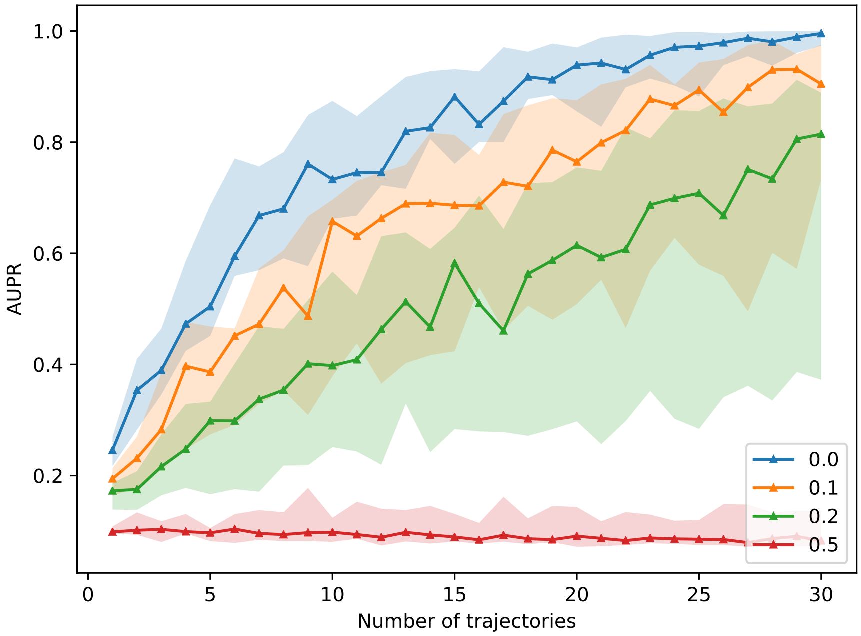

Figure D.2.: AUPR results over increasing number of samples for ψ_0-vs-rest. We plot the median with a line and the 25-75th percentile with the shaded area over 10 runs. The legend shows the value of error probability p_e. The performance deteriorates as the noise increases, however, with the increasing number of trajectories the metric approaches to 1.

Bibliography

[1] Burton W. Andrews, Tau-Mu Yi, and Pablo A. Iglesias. Optimal noise filtering in the chemotactic response of Escherichia coli. *PLoS Comput Biol*, 2(11):e154, 2006.

[2] Clive G. Bowsher and Peter S. Swain. Environmental sensing, information transfer, and cellular decision-making. *Current opinion in biotechnology*, 28:149–155, 2014.

[3] Kendrick Boyd, Kevin H. Eng, and C. David Page. Area under the precision-recall curve: point estimates and confidence intervals. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 451–466. Springer, 2013.

[4] Ido Cohn, Tal El-Hay, Nir Friedman, and Raz Kupferman. Mean field variational approximation for continuous-time Bayesian networks. *The Journal of Machine Learning Research*, 11:2745–2783, 2010.

[5] Arnaud Doucet and Adam M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

[6] Julien O Dubuis, Gašper Tkačik, Eric F Wieschaus, Thomas Gregor, and William Bialek. Positional information, in bits. *Proceedings of the National Academy of Sciences*, 110(41):16301–16308, 2013.

[7] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in neural information processing systems*, pages 2137–2145, 2016.

[8] Nigel R. Franks, Stephen C. Pratt, Eamonn B. Mallon, Nicholas F. Britton, and David J. T. Sumpter. Information flow, opinion polling and collective intelligence in house–hunting social insects. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 357(1427):1567–1583, 2002.

[9] Daniel T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of computational physics*, 22(4):403–434, 1976.

[10] Simon Godsill. Particle filtering: the first 25 years and beyond. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7760–7764. IEEE, 2019.

[11] Lirong Huang, Loïc Paulevé, Christoph Zechner, Michael Unger, Anders Hansen, and Heinz Koeppl. Supporting information for reconstructing dynamic molecular states from single-cell time series. 2016.

[12] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[13] John Kerekes. Receiver operating characteristic curve confidence intervals and regions. *IEEE Geoscience and Remote Sensing Letters*, 5(2):251–255, 2008.

[14] P. A. W. Lewis and Gerald S. Shedler. Simulation of nonhomogeneous Poisson processes by thinning. *Naval research logistics quarterly*, 26(3):403–413, 1979.

[15] Eric Libby, Theodore J. Perkins, and Peter S. Swain. Noisy information processing through transcriptional regulation. *Proceedings of the National Academy of Sciences*, 104(17):7151–7156, 2007.

[16] Kevin P. Murphy. A survey of POMDP solution techniques. *Technical Report, University of California at Berkeley*, 2000.

[17] Ranjit Nair, Milind Tambe, Makoto Yokoo, David Pynadath, and Stacy Marsella. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *IJCAI*, volume 3, pages 705–711, 2003.

[18] Uri Nodelman, Christian R. Shelton, and Daphne Koller. Continuous time Bayesian networks. In *Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence*, pages 378–387, 2002.

[19] Uri Nodelman, Christian R. Shelton, and Daphne Koller. Learning continuous time Bayesian networks. In *Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence*, pages 451–458, 2003.

[20] Yosihiko Ogata. On Lewis' simulation method for point processes. *IEEE transactions on information theory*, 27(1):23–31, 1981.

[21] Keith Paarporn, Ceyhun Eksin, and Joshua S. Weitz. Information sharing for a coordination game in fluctuating environments. *Journal of theoretical biology*, 454:376–385, 2018.

[22] Theodore J. Perkins and Peter S. Swain. Strategies for cellular decision-making. *Molecular systems biology*, 5(1):326, 2009.

[23] Marian-Andrei Rizoiu, Young Lee, Swapnil Mishra, and Lexing Xie. Hawkes processes for events in social media. In *Frontiers of Multimedia Research*, pages 191–218. 2017.

[24] Katsushige Sawaki and Akira Ichikawa. Optimal control for partially observable Markov decision processes over an infinite horizon. *Journal of the Operations Research Society of Japan*, 21(1):1–16, 1978.

[25] Vahid Shahrezaei and Peter S. Swain. The stochastic nature of biochemical networks. *Current opinion in biotechnology*, 19(4):369–374, 2008.

[26] Edward J. Sondik. The optimal control of partially observable Markov processes over the infinite horizon: Discounted costs. *Operations research*, 26(2):282–304, 1978.

[27] Lukas Studer, Loïc Paulevé, Christoph Zechner, Matthias Reumann, María Rodríguez Martínez, and Heinz Koeppl. Marginalized continuous time Bayesian networks for network reconstruction from incomplete observations. In *Proceedings of the 30th Conference on ArtificialIntelligence (AAAI 2016)*, pages 2051–2057, 2016.

[28] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: An introduction.* MIT press, 2018.

[29] Ying Tan and Zhong-yang Zheng. Research advance in swarm robotics. *Defence Technology*, 9(1):18–39, 2013.

[30] Sebastian Thrun. Monte Carlo POMDPs. In *Advances in neural information processing systems*, pages 1064–1070, 2000.

[31] Christopher M. Waters and Bonnie L. Bassler. Quorum sensing: cell-to-cell communication in bacteria. *Annual Review of Cell and Developmental Biology*, 21:319–346, 2005.

[32] C. C. White and David P. Harrington. Application of Jensen's inequality to adaptive suboptimal design. *Journal of Optimization Theory and Applications*, 32(1):89–99, 1980.