STATISTICAL VERIFICATION AND DIFFERENTIAL PRIVACY IN CYBER-PHYSICAL SYSTEMS

BY

YU WANG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mechanical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Doctoral Committee:

Professor Geir E. Dullerud, Chair
Professor Sayan Mitra
Professor Matthew West
Professor Magnus Egerstedt
Professor Mahesh Viswanathan

ABSTRACT

This thesis studies the statistical verification and differential privacy in Cyber-Physical Systems. The first part focuses on the statistical verification of stochastic hybrid system, a class of formal models for Cyber-Physical Systems. Model reduction techniques are performed on both Discrete-Time and Continuous-Time Stochastic Hybrid Systems to reduce them to Discrete-Time Markov Chains and Continuous-Time Markov Chains, respectively; and statistical verification algorithms are proposed to verify Linear Inequality LTL and Metric Interval Temporal Logic on these discrete probabilistic models. In addition, the advantage of stratified sampling in verifying Probabilistic Computation Tree Logic on Labeled Discrete-Time Markov Chains is studied; this method can potentially be extended to other statistical verification algorithms to reduce computational costs.

The second part focuses on the Differential Privacy in multi-agent systems that involve share information sharing to achieve overall control goals. A general formulation of the systems and a notion of Differential Privacy are proposed, and a trade-off between the Differential Privacy and the tracking performance of the systems is demonstrated. In addition, it is proved that there is a trade-off between Differential Privacy and the entropy of the unbiased estimator of the private data, and an optimal algorithm to achieve the best trade-off is given.

To my family and my advisor

ACKNOWLEDGMENTS

I would like to thank my advisor, Professor Geir E. Dullerud for guiding and supporting me over the past six years. You have set an example of excellence as a researcher, mentor, instructor, and role model. I would like to thank my thesis committee members Professor Sayan Mitra, Professor Matthew West, Professor Magnus Egerstedt, and Professor Mahesh Viswanathan for all of their guidance through this process; your discussion, ideas, and feedback have been absolutely invaluable. I'd like to thank my collaborators, Nima Roohi, Zhenqi Huang, and Matthew Hale, who contributed to this research. I am very grateful to all of you. Finally, I would like to thank my father Zhilin Wang, my mother Bing Yu, and my wife Jingjing Shi for the love, support, and constant encouragement I have gotten over the years. I undoubtedly could not have done this without you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

LTL	Linear Temporal Logic
CTL	Computation Tree Logic
iLTL	Linear Inequality Linear Temporal Logic
PCTL	Probabilistic Computation Tree Logic
MTL	Metric Temporal Logic
MITL	Metric Interval Temporal Logic
BA	Büchi Automata
TA	Timed Automata
DTMC	Discrete-Time Markov Chain
CTMC	Continuous-Time Markov Chain
CPS	Cyber-Physical System
SHS	Stochastic Hybrid System

LIST OF SYMBOLS

$\emptyset$	Empty set
$[n]$	Integer set $\{1, 2, \ldots, n\}$
$\mathbb{N}$,$\mathbb{Z}$	Set of natural numbers and integers
$\mathbb{Q}$, $\mathbb{Q}_{\geq 0}$	Set of rational and non-negative rational numbers
$\mathbb{R}$, $\mathbb{R}_{>0}$ and $\mathbb{R}_{\geq 0}$	Set of real, positive real, and non-negative real numbers
$\mathbb{I}_{\geq 0}$	Set of intervals on $\mathbb{Q}_{\geq 0}$.
$s^{(n)}$, $s^{(\infty)}$	Finite sequence $\{s_0, s_1, \ldots, s_n\}$, infinity sequence $\{s_0, s_1, \ldots\}$
M_{ij}	Entry in the i^{th} row, j^{th} column of matrix M
$S^{\boldsymbol{\omega}}$	Set of infinite sequences in S
$\lvert S\rvert$	Cardinality of set S
2^S	Power set of S
∂X	Boundary of set X
$\mathbb{P}$, $\mathbb{E}$	Probability and the expected value
$\lvert \cdot \rvert_p$	ℓ_p] Absolute value
$\lVert \cdot \rVert_p$	ℓ_p] Norm

CHAPTER 1

INTRODUCTION

Cyber-Physical Systems (CPS) model physical processes that are controlled or monitored by computer-based algorithms. They arise in various real-world applications ranging from automobiles [1], smart grids [2] and biology [3, 4, 5, 6, 7]. These systems typically involve discrete, continuous, and stochastic behaviors, as well as communication and information sharing.

In these contexts, it is often useful to determine if the systems meet their time-dependent design goals. However, the verification problem is computationally very challenging — even for systems with very simple dynamics that exhibit no stochasticity, and for the most basic class of safety properties, namely invariants, the problem of determining if a system meets its safety goals is undecidable [8]. The difficulty of the verification problem largely arises from the fact that the state space of such systems has uncountably many states.

The computational challenge posed by the verification problem is often addressed by constructing and then analyzing a simpler finite state model. The finite state model is typically an *abstraction* or a conservative over-approximation of the original system, *i.e.*, every behavior of the system is exhibited by the finite state model, but the finite state model may have additional behaviors that are not system behaviors. This approach has been used to verify [9, 10, 11] and design controllers [12, 13, 14, 15] for non-stochastic systems, and to verify [4, 5, 7, 16, 17] and design controllers [18] for stochastic hybrid systems. For such abstractions, if the finite state model is safe, then so is the original system. However, if the finite state model is unsafe, then not much can be concluded about the safety of the original system because the finite state model is an over-approximation.

In the first part of this thesis, a scalable approach is proposed to verification of stochastic hybrid systems that rely on constructing a finite state approximation that is "equivalent" to the original system. The advantage of using a reduction that is approximately equivalent to the original system is that analyzing the finite state model not only allows us to conclude the safety of the hybrid stochastic system, but also its non-safety. The finite state Markov chain reduction is constructed by using the Mori-Zwanzig model reduction method [19, 20].

In order to explain the relationship between the Markov chain and the stochastic hybrid system, it is useful to recall that there are two broad approaches to defining the semantics of a stochastic process. One approach is to view a stochastic system as defining a measure space on the collection of executions, namely a sequence of states that the system may possibly go through. The other approach is to view the stochastic system as defining a transformation on distributions; in such a view, the behavior of the stochastic model is captured by a sequence of distributions, starting from some initial distribution. It has been observed that with respect to the first semantics (of measures on executions) it is not possible to construct a finite state Markov chain that is "equivalent" to an infinite state system [7]. Here, in contrast, it is shown that the Mori-Zwanzig reduction method constructs a finite state Markov chain that is approximately equivalent to a stochastic hybrid system with respect to the second semantics. That is, the distribution on states of the Markov chain at any time instance is close to the distribution at the same time defined by the stochastic hybrid system.

This observation is similar in spirit to the results first established for non-stochastic, stable, hybrid systems [12, 21, 22], and later extended to stochastic dynamical systems [23, 24]. When compared to [23, 24], a more general class of stochastic hybrid systems that have multiple modes and jumps with guards and resets is considered here. Second, the reduced system is a Markov chain, whereas in [23, 24] the stochastic system is approximated by a finite state, non-stochastic model. Finally, the notion of distance between the stochastic hybrid system and the reduced system is slightly different.

The fact that the reduced Markov model is approximately equivalent to the original stochastic hybrid system is exploited to verify stochastic hybrid systems. Approximate equivalence ensures that analyzing the reduced model with respect to a suitably strengthened property determines whether the initial stochastic hybrid system meets or violates its requirements. Therefore, a scalable verification approach can be obtained by developing algorithms to verify finite state Markov chains.

Since the reduced system, even though having finite states, is likely to have a large number of states, a statistical approach is adopted for verification [25, 26, 27, 28, 29, 30] as opposed to a symbolic one. In statistical model checking, the model being verified is simulated multiple times, and the drawn simulations are analyzed to see if they constitute a statistical evidence for the correctness of the model. Statistical model checking algorithms have been developed for logics that reason about measures of executions [6, 25, 31, 32]. However, since the reduced Markov chain is only close to the stochastic hybrid system in a distributional sense, these algorithms are not applicable.

Instead, new statistical model checking algorithms are developed for temporal logics over discrete and continuous time that reason about sequences of distributions. It is believed that this approach to verifying stochastic hybrid systems is scalable, as an initial experimental evaluation supports this claim (see Example 1). Also, this approach is the first to succeed in verifying [33] a highly non-linear model including lookup tables of a powertrain control system that was proposed as a challenging problem for verification tools by Toyota engineers [1].

The end of the first part is devoted to demonstrating that the computational costs of statistical verification algorithms can be significantly reduced if the statistical model checker draws *correlated* samples, as opposed to independent samples in previous studies [34, 35, 36, 37, 38, 39, 40, 41].

A common way to generate such negatively correlated samples with negligible additional computational cost is *stratified sampling*, which has been popular among the statistics community in improving the accuracy of statistical estimation [42, 43]. The general idea is to partition the sample space into different cells and draw one sample from each one of them. The stratified samples are repellent to each other — a sample occupying some cell forbids other samples entering the cell. Therefore, the stratified samples will be negatively correlated.

The idea of using stratified samples in statistical model checkers is demonstrated by a statistical verification algorithm for checking finite horizon Probabilistic Computation Tree Logic (PCTL) properties on Discrete-Time Markov chains (DTMC) using stratified sampling. It is shown by theory and numerical experiments that this algorithm, based on a sequential probability ratio test that work with stratified samples, helps reduce the total number of samples (number of strata × number of blocks of stratified samples) needed for a statistical model checker to be confident in its answer.

Another challenging problem in the study of Cyber-Physical Systems is the Differential Privacy of communication in distributed setups. Here, data about the individual participating agents can help achieve better system-level performance; however, at the same time, it is a requirement that the private data of these individuals be protected. Examples include peak generation scheduling using consumption data obtained from smart electric meters [44], traffic-aware navigation based on location and destination data obtained from smart GPS [45, 46], and data aggregation from sensor networks [47, 48].

In the second part of this thesis, the trade-off between ε-differential privacy and performance is studied in the context of discrete-time linear distributed control systems. In these systems, N agents operate in a shared environment which couples their dynamics. The agents coordinate with each other by communicating via a central server, while tracking individual desired signals, referred to as *preferences*. The preferences together with the initial states are the *private data* of the individual agents, and can be inferred from intercepted communications if agents share their precise state.

To keep the sensitive data private, one common approach is to add noise to the communicated information. The effectiveness of such an approach can be measured by using the concept of ε-differential privacy which stemmed from the study of stochastic databases [49, 50, 51, 52] and was later extended to dynamical systems [53, 54]. It is a frequently used measure of privacy in various settings, such as optimization [55, 56, 57] and consensus [53, 58]. Roughly, ε-differential privacy ensures that the probability distribution of any observation does not change substantially with a change in the private data corresponding to any *one* agent.

During the past decade, several varieties of differential privacy have been proposed [51, 53, 59, 60, 61]. The definition of differential privacy used in this current work (Definition 22) is introduced in [61] which augmented the most common definition of differential privacy [51, 59] with metrics. The main technical adjustment to the common definition is the introduction of a notion distance on the continuous space of private data; a consequence of the generalization is that greater changes in the private data of an agent now permit greater differences between the corresponding probability distributions of observation sequences.

While there are several notions of data privacy in the computer science literature, the quantitative and statistical nature of differential privacy makes it suitable for adoption in control. The notion of differential privacy is first introduced in the context of statistical databases where agents' private information is their participation status in the database [49, 51]. In this context, two data sets are *adjacent* if they are different in the (binary) data corresponding to a single agent and are identical elsewhere. The definition of adjacency varies between contexts. For example, for real-valued databases, like the definitions presented in [62, 63], adjacent data sets are defined as identical data sets with one agent whose values are close (as measured by a metric on its real-valued variables). This notion of differential privacy guarantees that two sets of behaviors, starting from two adjacent initial states and corresponding to any output sequence, are statistically close. Various mechanisms for achieving differential privacy have been studied in the literature [64, 65, 66]. The Laplace mechanism requires adding a Laplace noise to the query output and was proposed in [49].

In [54, 60, 67], the authors develop a notion of differential privacy which ensures that a filter cannot precisely estimate the input to a dynamical system by looking at its output stream. Laplace and Gaussian mechanisms are presented for converting an ordinary dynamical system to a differentially private one and a Kalman filter is designed to estimate the states of a Gaussian mechanism with minimized ℓ_2 error. The sufficient condition of the minimization problem is established in the form of linear matrix inequalities. However, whether the Gaussian mechanism is the best mechanism (in terms of metric like ℓ_2-norm or entropy) is not addressed.

The problems introduced in these two papers differ in several ways from the one studied in this thesis. First, in the class of systems studied in this thesis, an agent's dynamics may be coupled with the environment which depends on the aggregate of all other agents' states. Secondly, these systems are "closed loop" and the noise added for privacy in one round affects all future states of the system. Further, in Section 8 an optimization problem is formulated for a general class of "one-shot" mechanisms and proved that the Laplace mechanism is optimal since it minimizes system entropy. This optimality result is then generalized to feedback dynamical systems.

When system noise is introduced, the quality of communication deteriorates, and thus the performance of the system will be negatively influenced. In [68], the authors study the optimal noise-adding mechanisms that minimize certain ℓ_1 cost function while keeping the query ε-differential privacy and demonstrate that the optimal solution is the staircase mechanism. In this thesis, the problem is studied in the background of distributed control systems where time evolves and therefore communication is repeated, in contrast to single-query problems. In addition, a stronger metric definition of ε-differential privacy is adopted (see Section 2.6). The performance measure is the aggregated mean-squared tracking deviation of the agent trajectories from their preferences and the (Shannon) entropy of the estimated private data.

One main contribution of this thesis is that an ε-differentially private mechanism communication strategy is designed for the discrete-time linear distributed control systems and a trade-off is established between ϵ-differential privacy and system performance. Specifically, the *cost of privacy*—namely the increase in the mean-squared error of the agent trajectories from their preferences—is shown to be $O(\frac{T^3}{N\epsilon^2})$ for stable systems, where T is the time-horizon length and N is the number of agents. This cost can grow exponentially with T for unstable systems.

Beneficially from a privacy perspective, system noise hinders the accurate estimation of the agents' states. The other main contribution of this thesis is that a trade-off is established between ε-differentially privacy and the accuracy of optimal estimation. Specifically, it is proved that the entropy of unbiased estimators of the private data has a lower bound given in terms of N, the number of agents, and n their individual-subsystem state dimension. A noise-adding mechanism that achieves this minimum bound is presented.

The rest of the thesis is organized as follows. The Preliminaries are given in Chapter 2: specifically, Sections 2.1 to 2.3 are devoted to various temporal logics in discrete or continuous time; Section 2.4 contains the essentials of statistical verification; Section 2.5 introduces the Continuous-Time and Discrete-Time Stochastic Hybrid Systems; and Section 2.6 is on the basics of differential privacy.

Part I of this thesis is devoted to the statistical verification of temporal logic based on my previous papers [27, 28, 29, 33, 69, 70]. In Chapter 3, a model reduction technique is demonstrated on the Discrete-Time and Continuous-Time Stochastic Hybrid Systems to reduce them to Discrete-Time Markov Chains and Continuous-Time Markov Chains, respectively. Following this, statistical verification algorithms using *independent* samples are proposed for checking Linear Inequality LTL on Discrete-Time Markov Chains and Metric Interval Temporal Logic on Continuous-Time Markov Chains in Chapter 4. Finally, in Chapter 5, the advantage of using *stratified* samples are illustrated by checking finite horizon Probabilistic Computation Tree Logic (PCTL) properties on Discrete-Time Markov Chains.

Part II of this thesis is devoted to the differential privacy in Multi-Agent Systems based on my previous papers [71, 72, 73, 74, 75]. In Chapter 6, the problem formulation of differential privacy in these systems are given. In Chapter 7, the trade-off between the level of differential private of the agents' private data and the tracking performance of the systems are studied. In Chapter 8, the impact of differential privacy on estimating the agents' private data is studied and an optimal mechanism of adding correlated noise is proposed to minimize the entropy of the unbiased estimators of the private data. The conclusion of this thesis is given in Chapter 9.

CHAPTER 2

PRELIMINARIES

2.1 Temporal Logics on Transition Systems

Temporal logics are sets of *syntax* and *semantics* rules to reason formally about events over time. It is used extensively to describe the transitional behavior of both discrete or continuous dynamical systems. In the discrete-time domain, Linear Temporal Logic (LTL) and Computational Tree Logic (CTL) are used to specify properties on the *paths* and *states* of a system, respectively. Verifying these two logics is *decidable* when the semantics are defined over Labeled Transition Systems (LTS). In the continuous-time domain, Metric Temporal Logic (MTL) extend Linear Temporal Logic. However, verifying Metric Temporal Logic is *undecidable*. Instead, only a fragment, Metric Interval Temporal Logic (MITL), also known as Signal Temporal Logic (STL) [76, 77, 78], is *decidable*. Temporal logic can also be extended to specify properties on probabilistic systems. For example, Probabilistic Computational Tree Logic (PCTL), extending Computational Tree Logic, can specify temporal properties on labeled Markov Chains (MC).

2.1.1 Labeled Transition Systems

Labeled Transition Systems, a.k.a. Kripke Structures is a nondeterministic discrete-time and finite-state dynamical system with each state labeled by a set of atomic propositions AP holding on that state.

Definition 1 (Labeled Transition Systems)**.** *A Labeled Transition System $\mathcal{T}$ is a tuple,* $(\mathsf{S}, \mathsf{AP}, \mathsf{T}, \mathsf{s}_{\mathrm{init}}, \mathsf{L})$*, consisting of*

- *a finite set of states* S*,*

- *a finite set of atomic propositions* AP*,*

- *a transition function* $\mathsf{T} : \mathsf{S} \to 2^{\mathsf{S}}$,

- *an initial state* s_{init},

- *a labeling function* $\mathsf{L} : \mathsf{S} \to 2^{\mathsf{AP}}$.

An infinite sequence of states $\mathsf{s}^{(\infty)} \subseteq \mathsf{S}$ *is called a* path *if*

1. $\mathsf{s}_0 = \mathsf{s}_{\text{init}}$,

2. $\mathsf{s}_{i+1} \in \mathsf{T}(\mathsf{s}_i)$ *for* $i = 0, 1, 2, \ldots$.

An infinite sequence of atomic propositions $\mathsf{a}^{(\infty)} \subseteq \mathsf{AP}$ *is called an* execution *if there exists a path* $\mathsf{s}^{(\infty)}$ *such that*

$$\mathsf{a}_i \in \mathsf{L}(\mathsf{s}_i) \textit{ for } i = 0, 1, 2, \ldots.$$

2.1.2 Linear Temporal Logic

Linear Temporal Logic is a set of formal rules for specifying *path* properties over time. It is composed of three elements: a set of atomic propositions AP whose correctness is known at each time instance, propositional logic operators $\{\neg, \wedge, \vee, \ldots\}$, and temporal operators $\{\mathbf{X}, \mathbf{U}, \ldots\}$.

Definition 2 (LTL Syntax)**.** *A Linear Temporal Logic formula is defined by*

$$\varphi ::= \mathsf{a} | \neg\varphi | \varphi \wedge \psi | \mathbf{X}\varphi | \varphi \mathbf{U}_T \psi,$$

where $\mathsf{a} \in \mathsf{AP}$ *is an atomic proposition and* $T \in \mathbb{N}$ *is the time bound. When* $T = \infty$*, write* $\mathbf{U}_\infty$ *as* $\mathbf{U}$.

In Definition 2, $\mathbf{X}\varphi$ stands for "next φ", namely, the property φ holds at the next time instance; $\varphi\mathbf{U}_T\psi$ stands for "φ until ψ no later than time T", namely, the property ψ holds at some time no later than T and before that φ holds. Also, a minimal set of logic operators is used in Definition 2 — additional temporal operators $\mathbf{R}, \mathbf{F}, \mathbf{G}$ are defined as follows:

- $\varphi\mathbf{R}_T\psi = \neg(\neg\varphi\mathbf{U}_T\neg\psi)$ stands for "φ release ψ no later than time T", namely, φ remains true before some time no later than T and then ψ becomes and remains true.

- $\mathbf{F}\varphi = \mathrm{True}\mathbf{U}\varphi$ stands for “final φ”, namely φ finally becomes true.

- $\mathbf{G}\varphi = \neg(\mathbf{F}\neg\varphi)$ stands for “global φ”, namely φ is always true.

The standard semantics of Linear Temporal Logic is defined on Labeled Transition Systems having the same set of atomic propositions.

Definition 3 (LTL Semantics)**.** *Let $\mathcal{T} = (\mathsf{S}, \mathsf{T}, \mathsf{s}_{\text{init}}, \mathsf{L})$ be a Labeled Transition System, and $\mathsf{s}^{(\infty)}$ be a path of $\mathcal{T}$. The satisfaction relation $\models$ is defined recursively by*

$$\begin{array}{ll}
\mathsf{s}^{(\infty)} \models \mathsf{a} & \textit{iff}\ \mathsf{a} \in \mathsf{L}(\mathsf{s}_0) \\
\mathsf{s}^{(\infty)} \models \neg\varphi & \textit{iff}\ \mathsf{s}^{(\infty)} \not\models \varphi \\
\mathsf{s}^{(\infty)} \models \varphi \wedge \psi & \textit{iff}\ \mathsf{s}^{(\infty)} \models \varphi \text{ and } \mathsf{s}^{(\infty)} \models \psi \\
\mathsf{s}^{(\infty)} \models \mathbf{X}\varphi & \textit{iff}\ \mathcal{S}\big(\mathsf{s}^{(\infty)}\big) \models \varphi \\
\mathsf{s}^{(\infty)} \models \varphi \mathbf{U}_T \psi) & \textit{iff}\ \exists t \leq T,\ \Big(\forall \tau < t,\ \mathcal{S}^{(\tau)}\big(\mathsf{s}^{(\infty)}\big) \models \varphi\Big) \wedge \mathcal{S}^{(t)}\big(\mathsf{s}^{(\infty)}\big) \models \psi
\end{array}$$

where $\mathcal{S}$ is the time shift operator and $\mathcal{S}^{(t)}$ is the t-fold composition.

2.1.3 Büchi Automata

A Büchi automaton is a discrete-time finite-state dynamical system with transitions between states labeled by alphabets. It takes a word, namely, a sequence of alphabets, as input and returns *accept* when the corresponding sequence of transitions lead to an accept state. The transitions can be either *deterministic* or *nondeterministic*, and nondeterminism gives extra expressive power. In this thesis, a Büchi automaton means a nondeterministic Büchi automaton.

Definition 4 (Büchi Automata)**.** *A non-deterministic Büchi automaton $\mathcal{B}$ is a tuple, $(\mathsf{S}, \mathsf{A}, \mathsf{T}, \mathsf{s}_{\text{init}}, \mathsf{S}_{\text{final}})$, consisting of*

- *a finite set of states S,*

- *a finite set of alphabet A,*

- *a transition function $\mathsf{T} : \mathsf{S} \times \mathsf{A} \to 2^{\mathsf{A}}$,*

- *an initial state s_{init},*

- *a set of accept states $\mathsf{S}_{\text{final}} \subseteq \mathsf{S}$.*

The automaton $\mathcal{B}$ accepts a word $\mathsf{a}^{(n)} \subseteq \mathsf{A}$*, if there exists an* accepting run $\mathsf{s}^{(n+1)} \subseteq \mathsf{S}$ *that satisfies*

1. $\mathsf{s}_0 = \mathsf{s}_{\text{init}}$*,*

2. $\mathsf{s}_{i+1} \in \mathsf{T}(\mathsf{s}_i, \mathsf{a}_i)$ *for* $i = 0, 1, \ldots, n$*,*

3. $\inf(\mathsf{s}^{(n)}) \in \mathsf{S}_{\text{final}}$*.*

A sequence of states satisfying (1) and (2) is called a run*. The set of words accepted by the automaton $\mathcal{B}$ is called the* language *accepted by $\mathcal{B}$ and is denoted by* $\mathrm{Lang}(\mathcal{B})$*.*

For any LTL formula φ, a Büchi automaton $\mathcal{B}_\varphi$ can be constructed such that $\mathtt{Lang}(\mathcal{B}_\varphi) = [\![\varphi]\!]$, *i.e.*, the set of infinite words that satisfy φ is exactly those that are accepted by $\mathcal{B}_\varphi$ [79, 80, 81].

2.1.4 Computational Tree Logic

Computational Tree Logic is a set of formal rules for specifying *state* properties over time. In addition to atomic propositions, propositional logic operators and temporal operators, it contains two extra quantifiers $\mathbf{A}, \mathbf{E}$.

Definition 5 (CTL Syntax)**.** *A Computation Tree Logic* state *formula is defined by*

$$\Phi ::= \mathsf{a} | \neg\Phi | \Phi \wedge \Psi | \mathbf{A}\varphi | \mathbf{E}\varphi,$$

where $\mathsf{a} \in \mathsf{AP}$ *is an atomic proposition and φ is a path formula; and a Computation Tree Logic* path *formula is defined by*

$$\varphi ::= \mathbf{X}\Phi | \Phi \mathbf{U}_T \Psi,$$

where Φ, Ψ are state formulas and $T \in \mathbb{N}$ is the time bound. When $T = \infty$, write $\mathbf{U}_\infty$ as $\mathbf{U}$.

In most cases, Computation Tree Logic *state* formulas are of interest. Although Computation Tree Logic contains more logic operators than Linear Temporal Logic, the two logics are not comparable — there are properties that can be expressed by one but not the other. This is because Computation Tree Logic does not allow successively nested temporal operators — a temporal operator $\mathbf{X}$

or $\mathbf{U}$ have to be combined with a quantifier $\mathbf{A}$ or $\mathbf{E}$ before nesting with another temporal operator.

In Definition 5, the state formula $\mathbf{A}\varphi$ stands for "all φ", namely, all the paths starting from the state satisfy φ; and the state formula $\mathbf{E}\varphi$ stands for "exist φ", namely, there exist a path starting from the state satisfy φ Again, a minimal set of logic operators are used in Definition 2 — additional temporal operators $\mathbf{R}, \mathbf{F}, \mathbf{G}$ can be defined in the same way as in Section 2.1.2.

The standard semantics of Linear Temporal Logic is also defined on Labeled Transition Systems having the same set of atomic propositions.

Definition 6 (CTL Semantics)**.** *Let $\mathcal{T} = (\mathsf{S}, \mathsf{T}, \mathsf{s}_{\text{init}}, \mathsf{L})$ be a Labeled Transition System, and s_0 be a state of $\mathcal{T}$. For the state formulas, the satisfaction relation $\models$ is defined by*

$$\begin{array}{ll}
\mathsf{s}_0 \models \mathsf{a} & \textit{iff}\ \mathsf{a} \in \mathsf{L}(\mathsf{s}_0) \\
\mathsf{s}_0 \models \neg\Phi & \textit{iff}\ \mathsf{s}_0 \not\models \Phi \\
\mathsf{s}_0 \models \Phi \wedge \Psi & \textit{iff}\ \mathsf{s}_0 \models \Phi \text{ and } \mathsf{s}_0 \models \Psi \\
\mathsf{s}_0 \models \mathbf{E}\varphi & \textit{iff for some path}\ \mathsf{s}^{(\infty)} \models \varphi \\
\mathsf{s}_0 \models \mathbf{A}\varphi & \textit{iff for every path}\ \mathsf{s}^{(\infty)} \models \varphi
\end{array}$$

where the path $\mathsf{s}^{(\infty)}$ starts from s_0.

The satisfaction relation $\models$ for the path formulas is defined in the same way as Linear Temporal Logic in Definition 3.

2.2 Temporal Logic on Discrete-Time Markov Chains

Extending Discrete-Time Temporal Logics like Linear Temporal Logic and Computation Tree Logic on Probablistic Systems like Discrete-Time Markov Chains involves two steps:

1. Augmenting the syntax with the ability to reason about probability;

2. Transplanting the semantics of the logic operators to the Discrete-Time Markov Chains.

For Linear Temporal Logic, re-defining and re-interpreting the atomic propositions as inequalities about linear functionals of the probability distributions on the states of the Discrete-Time Markov Chains leads to a probabilistic extension called Linear Inequality LTL. For Computation Tree Logic, replacing the quantifiers $\mathbf{A}, \mathbf{E}$ with probabilistic quantifiers $\mathbf{P}$ leads to a probabilistic extension called Probabilistic Computation Tree Logic. These two extensions have the ability to reason on the temporal behavior of (labeled) Discrete-Time Markov Chains on the *distribution* and *state* level, respectively.

2.2.1 Discrete-Time Markov Chain

Definition 7 ((Labeled) Discrete-Time Markov Chains)**.** *A Discrete-Time Markov Chains is a tuple,* $(\mathsf{S}, \mathsf{T}, \mathsf{s}_{\text{init}})$*, consisting of*

- *a finite set of states* S*,*
- *a transition probability function* $\mathsf{T} : \mathsf{S} \times \mathsf{S} \to [0, 1]$*,*
- *an initial state* s_{init}*,*

such that

$$\sum_{\mathsf{s} \in \mathsf{S}} \mathsf{T}(\mathsf{s}, \mathsf{s}') = 1.$$

A Labeled Discrete-Time Markov Chain is a tuple $(\mathsf{S}, \mathsf{AP}, \mathsf{T}, \mathsf{s}_{\text{init}}, \mathsf{L})$ *augmenting the Discrete-Time Markov Chain by*

- *a finite set of atomic propositions* AP*,*
- *a labeling function* $\mathsf{L} : \mathsf{S} \to 2^{\mathsf{AP}}$*,*

An infinite sequence of states $\mathsf{s}^{(\infty)} \subseteq \mathsf{S}$ *is called a* path *if*

1. $\mathsf{s}_0 = \mathsf{s}_{\text{init}}$*,*
2. $\mathsf{T}(\mathsf{s}_i, \mathsf{s}_{i+1}) > 0$ *for* $i = 0, 1, 2, \ldots$*.*

An infinite sequence of atomic propositions $\mathsf{a}^{(\infty)} \subseteq \mathsf{AP}$ *is called an* execution *if there exists a path* $\mathsf{s}^{(\infty)}$ *such that*

$$\mathsf{a}_i \in \mathsf{L}(\mathsf{s}_i) \textit{ for } i = 0, 1, 2, \ldots.$$

Labeled Discrete-Time Markov Chains extends the common Discrete-Time Markov Chains by labeling the set of atomic propositions AP holding on each state. They can be viewed as an extension of Labeled Transition Systems by assigning a transition probability to each transition.

2.2.2 Linear Inequality LTL

A Probabilistic System produces a sequence of distributions $p^{(\infty)}$ over time. Linear Inequality LTL (iLTL) formulas reason about this sequence of distributions by connecting atomic propositions that are inequalities about linear functionals on the distributions with Linear Temporal Logic operators.

Definition 8 (iLTL Syntax)**.** *A Linear Inequality LTL formula is defined by*

$$\begin{aligned} \varphi \quad & ::= \text{ineq}|\neg\varphi|\varphi \wedge \psi|\mathbf{X}\varphi|\varphi\mathbf{U}_T\psi, \\ \text{ineq} \quad & ::= w(p) > c. \end{aligned}$$

where $c \in \mathbb{R}$ and w is a linear functional of distributions. By measure theory, for a probabilistic distribution p *on the states, it can be written as integration against a weight function $w : \mathsf{s} \to \mathbb{R}$ (with slightly abusing the notations)*

$$w(p) = \sum_{\mathsf{s} \in \mathsf{S}} w(\mathsf{s})p(\mathsf{s}).$$

The semantics of Linear Inequality LTL can be defined abstractly over a sequence of distributions $p^{(\infty)}$.

Definition 9 (iLTL Semantics)**.** *The semantics of Linear Inequality LTL is the same as Linear Temporal Logic in Definition 3 except that the atomic propositions are interpreted as*

$$\mathsf{s}^{(\infty)} \models \text{ineq } \textit{iff } w(p_0) > c + \epsilon \textit{ for some } \epsilon > 0$$

In Definition 8 and Definition 9, a minimal set of operators are used — additional operators can be defined, for example, $w(p) \leq c$ is equivalent to $\neg(w(p) > c)$. Since the semantics is defined in a robust manner, $w(p) \leq c$ and $w(p) \geq c$ are equivalent to $w(p) < c$ and $w(p) > c$.

2.2.3 Probabilistic Computation Tree Logic

Probabilistic Computation Tree Logic derives from Computation Tree Logic by replacing the quantifiers $\mathbf{A}, \mathbf{E}$ with probabilistic quantifiers $\mathbf{P}$.

Definition 10 (PCTL Syntax)**.** *Let* AP *be a set of atomic propositions. A PCTL formula is defined by*

$$\varphi ::= a|\neg\varphi|\varphi_1 \wedge \varphi_2|\mathbf{P}_J(\mathbf{X}\varphi)|\mathbf{P}_J(\varphi\mathbf{U}_T\psi),$$

where $a \in \mathsf{AP}$, $J \in [0,1]$ *is an interval with rational bounds* , $T \in \mathbb{N}$ *is a time bound. When* $T = \infty$*, write* $\mathbf{U}_\infty$ *as* $\mathbf{U}$.

With probabilistic modification, Probabilistic Computation Tree Logic has the ability to reason over the paths that start from a state in Labeled Discrete-Time Markov Chains.

Definition 11 (PCTL Semantics)**.** *Let* $(\mathsf{S}, \mathsf{AP}, \mathsf{T}, \mathsf{s}_{\text{init}}, \mathsf{L})$ *be a Labeled Discrete-Time Markov Chain. The semantics of PCTL is defined recursively by*

$$\begin{array}{ll}
s_0 \models a & \textit{iff } a \in L(s) \\
s_0 \models \neg\varphi & \textit{iff } s \not\models \varphi \\
s_0 \models \varphi \wedge \psi & \textit{iff } s_0 \models \varphi \text{ and } s_0 \models \psi \\
s_0 \models \mathbf{P}_J(\mathbf{X}\varphi) & \textit{iff } \mathbb{P}\big[\mathsf{s}^{(\infty)} \mid \mathcal{S}\big(\mathsf{s}^{(\infty)}\big) \models \varphi\big] \in J \\
s_0 \models \mathbf{P}_J(\varphi\mathbf{U}_T\psi) & \textit{iff } \mathbb{P}\big[\mathsf{s}^{(\infty)} \mid \exists t \leq T, \big(\forall \tau < t, \mathcal{S}^{(\tau)}\big(\mathsf{s}^{(\infty)}\big) \models \varphi\big) \wedge \mathcal{S}^{(t)}\big(\mathsf{s}^{(\infty)}\big) \models \psi\big] \in J
\end{array}$$

where $\mathcal{S}$ *is the time shift operator and* $\mathcal{S}^{(t)}$ *is the* t*-fold composition.*

It should be noted that Probabilistic Computation Tree Logic is not an extension of Computation Tree Logic. They are not comparable — there are properties that can be expressed by one but not the other. This is because, "for all" and "exists" are not equivalent to "happens with probability 1" and "happens with positive probability" in Probabilistic Systems.

2.3 Temporal Logic on Continuous Time Markov Chains

In the continuous time domain, Discrete-Time Markov Chains and Linear Inequality LTL are extended to Continuous-Time Markov Chains and a variation of Metric Interval Temporal Logic.

2.3.1 Continuous-Time Markov Chain

Definition 12 (Continuous-Time Markov Chains)**.** *A Continuous-Time Markov Chains is a tuple,* $(\mathsf{S}, \mathsf{T}, \mathsf{s}_{\text{init}})$*, consisting of*

- *a finite set of states* S*,*
- *a transition rate function* $\mathsf{T} : \mathsf{S} \times \mathsf{S} \to \mathbb{R}$*,*
- *an initial state* s_{init}*,*

such that

$$\sum_{\mathsf{s} \in \mathsf{S}} \mathsf{T}(\mathsf{s}, \mathsf{s}') = 0.$$

2.3.2 Metric (Interval) Temporal Logic

Metric Temporal Logic extends Linear Temporal Logic to the continuous time domain by extending and unifying the temporal operators $\mathbf{U}_T$, $\mathbf{X}$ to $\mathbf{U}_I$ where $I \in \mathbb{I}_{\geq 0}$ is an interval on $\mathbb{Q}_{\geq 0}$. Let $p(t)$ be the time-dependent distribution generated by a Continuous-Time Markov Chain. Similar to Section Section 2.2.2, to define a proper semantic of Metric Interval Temporal Logic, consider atomic propositions of the form $w(p) > c$ that are inequalities about linear functionals w of $p(t)$. Then the value of the linear functional $f(t) = w(p(t))$ is a real-valued function of time, commonly referred to as a *signal*.

Definition 13 (MITL Syntax)**.** *An Metric Temporal Logic formula is defined by*

$$\varphi = \mathsf{a} | \varphi \wedge \varphi | \varphi \vee \varphi | \varphi \mathbf{U}_I \varphi$$

where

$$\mathsf{a} ::= f > c \in \mathsf{AP}$$

and $I \in \mathbb{I}_{\geq 0}$*. The logic is called Metric Interval Temporal Logic or Signal Temporal Logic, if* $I \neq \emptyset$*.*

The semantics of Metric (Interval) Temporal Logic is defined with respect to the *signals*.

Definition 14 (MITL Semantics)**.** *Let φ be an MITL formula and f be a signal $f : \mathbb{R}_{\geq 0} \to 2^{\mathsf{AP}}$. The satisfaction relation $\models$between f and φ is defined by*

$$\begin{array}{lll} \mathsf{s}^{(\infty)} \models \text{ineq} & \textit{iff} & f(0) > c + \epsilon \textit{ for some } \epsilon > 0 \\ f(t) \models \neg\varphi & \textit{iff} & f(t) \not\models \varphi \\ f(t) \models \varphi \wedge \psi & \textit{iff} & (f(t) \models \varphi) \wedge (f(t) \models \psi) \\ f(t) \models \varphi \mathbf{U}_I \psi & \textit{iff} & \exists t \in I, (f(t) \models \psi) \\ & & \wedge \forall \tau \in (0, t), f(\tau) \models \varphi \end{array}$$

Similar to Section 2.2.2, in Definition 13 and Definition 14, a minimal set of operators are used — additional operators can be defined, for example, $f \leq c$ is equivalent to $\neg(f > c)$. Since the semantics is defined in a robust manner, $f \leq c$ and $f \geq c$ are equivalent to $f < c$ and $f > c$.

2.3.3 Timed Automata

Satisfiability and model checking problems for MITL with *abstract* atomic propositions are known to be EXPSPACE-complete [82]. The corresponding decision procedure has a close connection with timed automata.

Definition 15 (Timed Automata [83])**.** *A Timed Automaton $\mathcal{T}$ is a tuple* $(\mathsf{S}, \mathsf{C}, \mathsf{A}, \mathsf{L}, \mathsf{I}, \mathtt{E}, \mathsf{s}_{\text{init}}, \mathsf{S}_{\text{final}})$ *where*

- S *is a finite set of states.*
- C *is a finite set of clocks.*
- A *is a finite alphabet.*
- $\mathsf{L} \in \mathsf{S} \to \mathsf{A}$ *maps each location to the label of that location.*
- $\mathsf{I} \in \mathsf{S} \to (\mathsf{C} \to \mathbb{I}_{\geq 0})$ *maps each location to its invariant which is the set of possible values of variables in that location.*
- $\mathtt{E} \subseteq \mathsf{S} \times \mathsf{S} \times 2^{\mathsf{C}}$ *is a finite set of edges of the form (s, d, j), where $s = \mathtt{S}e$ is source of the edge; $d = \mathtt{D}e$ is destination of the edge; and $j = \mathtt{J}e$ is the set of clocks that are reset by the edge.*
- $\mathsf{s}_{\text{init}} \subseteq \mathsf{S}$ *is the set of initial locations.*
- $\mathsf{S}_{\text{final}} \subseteq \mathsf{S}$ *is the set of final locations.*

A *run* of the automaton $\mathcal{T}$ is a sequence of tuples $(\rho, \tau, \eta) \in \mathsf{S} \times \mathbb{I}_{\geq 0} \times \mathtt{E}$ with the following conditions holds: (i) $\rho_0 \in \mathsf{s}_{\text{init}}$, *i.e.*, ρ starts from an initial location s_{init}; (ii) $(\mathtt{S}\eta_n = \rho_n) \wedge (\mathtt{D}\eta_n = \rho_{n+1})$, *i.e.*, the source and destination of edges η_n are ρ_n and ρ_{n+1}; (iii) $\tau_0, \tau_1, \ldots$ is an ordered and disjoint partition of the time horizon $\mathbb{R}_{\geq 0}$; and (iv) $\forall t \in \tau_n, x \in \mathsf{C}$, $\varrho_n(x) + t - \underline{\tau}_n \in \mathsf{I}(\varrho_n, x)$, where $\varrho_{n+1}(x)$ is defined inductively by

$$\varrho_{n+1}(x) = \begin{cases} 0, & \text{if } x \in \mathtt{J}\eta_n \\ \varrho_n(x) + \overline{\tau}_n - \underline{\tau}_n, & \text{otherwise} \end{cases}$$

i.e., the clock times must satisfy the invariant of the current location. Here, $\underline{\tau}$ and $\overline{\tau}$ are the lower and upper bound of the interval.

A run satisfying the condition $\mathtt{inf}(\rho) \cap \mathsf{S}_{\text{final}} \neq \emptyset$, *i.e.*, some location from $\mathsf{S}_{\text{final}}$ has been visited infinitely many times by ρ, is called an *accepting run* of $\mathcal{T}$. Note that every run of $\mathcal{T}$ induces a function f of type $\mathbb{R}_{\geq 0} \to \mathsf{A}$ that maps t to $\mathsf{L}(\rho_n)$, where n is uniquely determined by the condition $t \in \tau_n$. The *language* of $\mathcal{T}$, denoted by $\mathtt{Lang}(A)$, is the set of all functions that are induced by accepting runs of $\mathcal{T}$.

For any MITL formula φ, a Timed Automaton $\mathcal{T}_\varphi$ can be constructed such that $\mathtt{Lang}(\mathcal{T}_\varphi) = [\![\varphi]\!]$, *i.e.*, the set of functions that satisfy φ is exactly those that are induced by accepting runs of $\mathcal{T}_\varphi$.

2.4 Statistical Verification

Generally, there are two approaches to verify temporal logic formulas on probabilistic systems: symbolic and stochastic. While the symbolic approach computes the exact satisfying probabilities of the objective temporal properties, the stochastic approach estimates these probabilities from samples with probabilistic guarantees. Compared to the symbolic approach, statistical verification algorithms have the following advantages: good scalability in high-dimension and complex system and applicability to “black-box” systems with unknown or inaccurate models.

2.4.1 Statistical Verification via Hypothesis Testing

The key step in statistical verification is to estimate the satisfying probability p of Linear Temporal Logic formula φ, namely a Probabilistic Computation Tree Logic formula $\mathbf{P}_{\sim p}\varphi$, from samples. More general temporal logic formulas can be verified by composition [36, 37, 38].

Let $X_1, X_2, \ldots$ be independent the sample paths. The correctness of φ can be determined for any trajectory X generated by the Markov chain $\mathcal{M}$. Define with a slight abuse of notation that

$$\varphi(X) = \begin{cases} 1, & \text{if } X \text{ satisfies } \varphi, \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

Consequently, checking $\mathbf{P}_{<p}\varphi$ is equivalent to a composite hypothesis testing problem

$$\begin{aligned} H_0 &: \mathbb{P}\left[\varphi(X)\right] < p, \\ H_1 &: \mathbb{P}\left[\varphi(X)\right] \geq p. \end{aligned} \tag{2.2}$$

Due to the robustness semantics Definitions 9 and 14, we assume that $|\mathbb{P}\left[\varphi(X)\right] - p| > \delta$ for some $\delta > 0$.

Assumption 1. *Let $|\mathbb{P}\left[\varphi(X)\right] - p| > \delta$ for some known indifference parameter $\delta > 0$. The interval $(\mathbb{P}\left[\varphi(X)\right] - \delta, \mathbb{P}\left[\varphi(X)\right] + \delta)$ is called the indifference Region.*

With Assumption 1, the composite hypothesis testing problem can be simplified to a simple hypothesis testing problem by testing the worst cases in the two hypothesis H_0 and H_1,

$$\begin{aligned} H_0' &: \mathbb{P}\left[\varphi(X)\right] \leq p - \delta, \\ H_1' &: \mathbb{P}\left[\varphi(X)\right] \geq p + \delta. \end{aligned} \tag{2.3}$$

The hypothesis testing problem (2.3) can be solved efficiently with a sequential probability ratio test (SPRT) as shown in [36, 37, 38]. Specifically, for a confidence level of type I error

$$\alpha = \mathbb{P}\left[\text{choose } H_1' | \mathbb{P}\left[\varphi(X)\right] = p - \delta\right] > 0, \tag{2.4}$$

and type II error

$$\beta = \mathbb{P}\left[\text{choose } H_0' | \mathbb{P}\left[\varphi(X)\right] = p + \delta\right] > 0, \tag{2.5}$$

consider the probability ratio

$$\Lambda(X^{(n)}) = \Pi_{i=1}^{n} \frac{(p+\delta)^{\varphi(X_i)}(1-p-\delta)^{1-\varphi(X_i)}}{(p-\delta)^{\varphi(X_i)}(1-p+\delta)^{1-\varphi(X_i)}}, \tag{2.6}$$

where $X^{(n)} = (X_1, \ldots, X_n)$. H_0 is accepted if $\Lambda(X^{(n)}) > \frac{\beta}{1-\alpha}$; H_0 is accepted if $\Lambda(X^{(n)}) > \frac{1-\beta}{\alpha}$; otherwise, draw a new sample X_{n+1}.

2.4.2 Stratified Sampling

Most statistical verification algorithms depend on independent samples. However, the verification cost can be significantly reduced if the statistical model checker draws *correlated* samples Let us consider the core task of a statistical model checker, namely, to determine if the measure of executions satisfying a property φ is greater than some threshold p. For simplicity, assume that the truth of φ itself can be determined by a finite prefix of the execution. In such a situation, the model checker draws sample executions, determines how many of the executions satisfy φ, and uses this to estimate the measure of paths satisfying φ. Thus, each sample can be viewed as a $0/1$-valued random variable X_i (which takes value 1 if the execution satisfies φ, and 0 otherwise), whose expectation is estimated by

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i.$$

One factor that plays an important role in determining how many samples are needed for the algorithm to be confident in its answer is the variance. Informally, the lower the variance of the estimate, the more likely the estimate is to be close to the actual mean, and therefore, the algorithm requires fewer samples. In general, the variance of the estimate is given by

$$\text{Var}\left[\bar{X}\right] = \frac{1}{n^2}\sum_{i=1}^{n} \text{Var}\left[X_i\right] + \frac{2}{n^2}\sum_{i=1}^{n}\sum_{j=i+1}^{n} \text{Cov}\left[X_i, X_j\right].$$

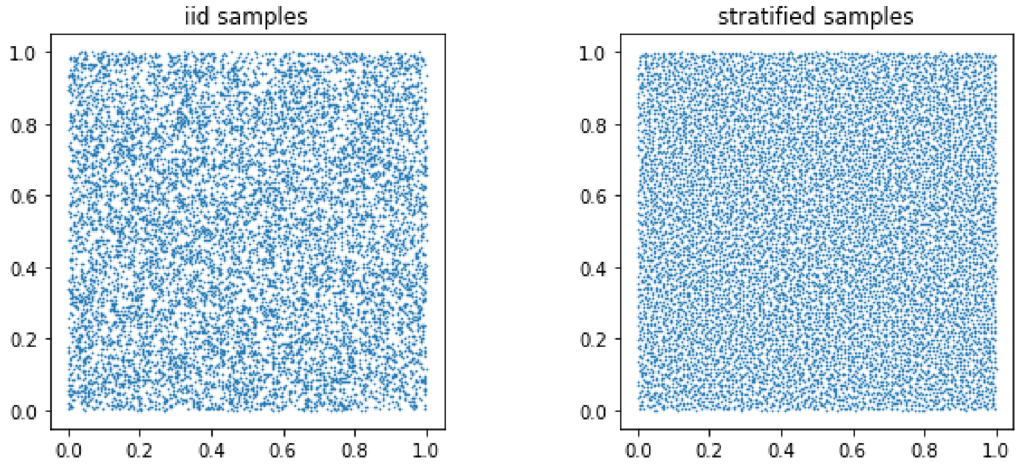


Figure 2.1: Independent samples v.s. stratified samples on unit square

If the samples are i.i.d., then the covariance is 0, and the variance is given by

$$\text{Var}\left[\bar{X}\right] = \frac{1}{n}\text{Var}\left[X\right].$$

However, as can be seen from the above expression, the variance can be reduced if the samples are *negatively correlated*, *i.e.*,

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n} \text{Cov}\left[X_i, X_j\right] \leq 0.$$

A common way to generate such negatively correlated samples with negligible additional computational cost is *stratified sampling*, which has been popular among the statistics community in improving the accuracy of statistical estimation [42, 43]. The general idea is to partition the sample space into different cells and draw one sample from each one of them. The stratified samples are repellent to each other — a sample occupying some cell forbids other samples entering the cell. Therefore, the stratified samples will be negatively correlated.

For example, $10\,000$ stratified samples can be drawn uniformly from the unit square $[0, 1]^2$ by first partitioning the area into 100×100 small cells, each of size 0.01×0.01, and then draw exactly one sample from each cell. Figure 2.1 shows graphically that compared to $10\,000$ independent samples, $10\,000$ stratified samples are negatively correlated, hence distribute more evenly on $[0, 1]^2$.

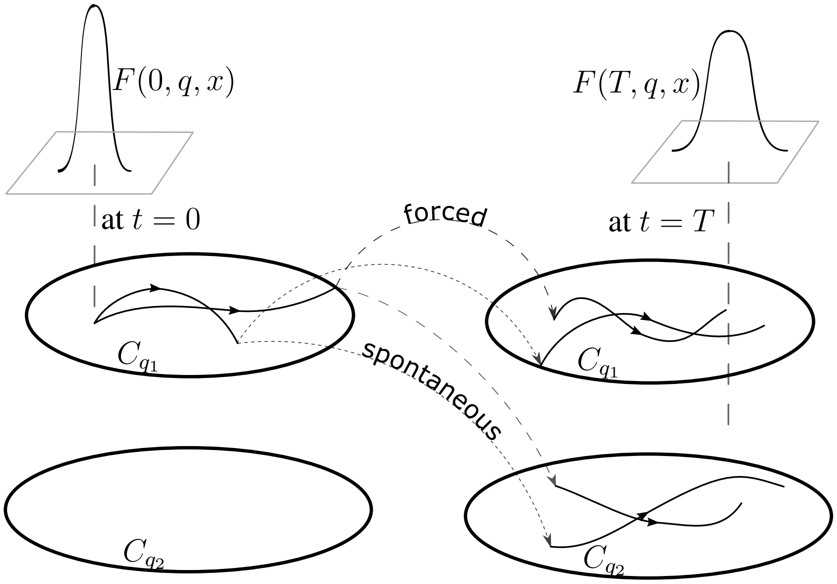


Figure 2.2: A continuous-time stochastic hybrid system with two discrete states at time 0 and T.

2.5 Stochastic Hybrid Systems

2.5.1 Continuous-Time

Continuous-time stochastic hybrid systems [84, 85, 86, 87] are a class of formal models for Cyber-physical systems that incorporates discrete spontaneous and forced jumps and continuous evolution and diffusion, as shown in Fig. 2.2. As a continuous-time probabilistic model, it has a Fokker-Planck formulation and interpretation.

The continuous and discrete states of the systems are denoted by $x \in \mathbb{R}^d$ and $q \in \mathcal{Q}$ respectively, where $\mathcal{Q} = \{q_1, \ldots, q_m\}$ is a finite set. The combination (q, x) is called the state of the system, and the product set $\mathbb{X} = \mathcal{Q} \times \mathbb{R}^d$ the state space.

The state space $\mathbb{X}$ of the system is divided into two regions: a flow set $\mathbb{A}$ and a jump set $\mathbb{B} = \mathbb{X} \backslash \mathbb{A}$. Define $\mathbb{A}_q = \left\{x \in \mathbb{R}^d \mid (q, x) \in \mathbb{A}\right\}$, and $\mathbb{B}_q$ similarly. It is assumed that each $\mathbb{A}_q$ is compact, and the boundaries $\partial\mathbb{A}_q$ are second-order continuously differentiable in x. On the flow set, the state x of the system evolves by a stochastic differential equation

$$\mathrm{d}\mathbf{x} = f(\mathbf{q}, \mathbf{x})\mathrm{d}t + g(\mathbf{q}, \mathbf{x})\mathrm{d}B_t, \tag{2.7}$$

where $\mathbf{q}$ and $\mathbf{x}$ are random processes describing the stochastic evolution of the continuous and discrete states, and B_t is the standard n-dimensional Brownian motion. The vector-valued function f specifies the drift of the state, and the matrix-valued function g describes the intensity of the diffusion [88, 89]. In (2.7), it is assumed that $f(q, \cdot)$ and $g(q, \cdot)$ are locally Lipschitz continuous. Meanwhile, the system jumps spontaneously by a non-negative integrable rate function $r_{\mathbb{A}}(q, x)$. The probability distribution of the jumping target is given by a non-negative integrable target distribution $h_{\mathbb{A}}(q', x', q, x)$. When the state of the system falls onto the jump set $\mathbb{B}$, the system is forced to jump. The probability distribution of the jumping target is given by a non-negative integrable target distribution $h_{\mathbb{B}}(q', x', q, x)$. The two target distributions $h_{\mathbb{A}}$ and $h_{\mathbb{B}}$ defined on two disjoint sets $\mathbb{A}$ and $\mathbb{B}$ are combined into one target transition h defined on the state space $\mathbb{X}$ of the system and satisfying

$$\sum_{q \in Q} \int_{\mathbb{R}^d} h(q', x', q, x) \mathrm{d}x' = 1. \tag{2.8}$$

The probability distribution $F(t, q, x)$ of the state of the system is determined by the standard Fokker-Planck equation

$$\begin{aligned} \frac{\partial F(t,q,x)}{\partial t} &= L(F(t,q,x)) \\ &= \underbrace{-\sum_{a=1}^{d} \frac{\partial}{\partial x_a}(f_a(q,x)F(t,q,x))}_{\text{drift}} \\ &\quad \underbrace{+\sum_{a=1}^{d}\sum_{b=1}^{d} \frac{\partial^2}{\partial x_a \partial x_b} \sum_{c=1}^{d} \frac{g_{ac}(q,x)g_{cb}(q,x)F(t,q,x)}{2}}_{\text{diffusion}} \\ &\quad \underbrace{-r(q,x)F(t,q,x)}_{\text{jump-out}} \\ &\quad \underbrace{+\sum_{q \in Q} \int_{\mathbb{R}^d} h(q,x,q',x')r(q',x')F(t,q',x')\mathrm{d}x,}_{\text{jump-in}} \end{aligned} \tag{2.9}$$

where L is the Fokker-Planck operator for the system. One can write symbolically that $F(t, q, x) = e^{tL}F(0, q, x)$. In (2.9), the four terms on the right hand side describe “drift”, “diffusion”, “jump-out” and “jump-in”, respectively.

On the other hand, a Fokker-Planck equation with proper boundary conditions that give unique solution defines a stochastic differential equation with jump and diffusion [88, 89]. Therefore, the following assumption is made.

Assumption 2. *It is assumed that the stochastic hybrid system given in this section is well defined in the sense that it gives a Fokker-Planck equation with a unique solution.*

A key component of the Mori-Zwanzig model reduction method is the invariant distribution. It is assumed that the continuous-time stochastic hybrid system has an invariant distribution with probability distribution function $F_{\text{inv}}(q,x)$ such that

$$L(F_{\text{inv}}(q,x)) = 0. \tag{2.10}$$

And, for any initial state, the probability distribution function $F(t,q,x)$ converges to the invariant distribution function $F_{\text{inv}}(q,x)$.

In many applications, the state of the system is only partially observable. Here, the observables of interest are given by

$$\begin{aligned} y(t) &= \mathbb{E}[y(q(t),x(t))] \\ &= \sum_{q\in\mathcal{Q}} \int_{\mathbb{R}^d} \gamma(q,x)F(t,q,x)\mathrm{d}x, \end{aligned} \tag{2.11}$$

where $\gamma(q,x)$ is a weight function on $\mathbb{X}$, which is integrable in x for each $q \in \mathcal{Q}$.

Example 1. *Throughout this part, the following example is used to illustrate the theorems. Consider a continuous-time stochastic hybrid system with two discrete states on* $\mathbb{X} = \{1\} \times [0,1] \cup \{2\} \times [2,4]$. *It reflects at* $x = 0$ *and* $x = 4$*, jumps uniformly to* $[2,4]$ *when hitting* $x = 1$*, and jumps uniformly to* $[1,2]$ *when hitting* $x = 2$*. It can jump spontaneously at any* $x \in \mathbb{X}$ *with rate* 1 *with* $r_{\mathbb{A}} = \mathbb{I}_{\mathbb{X}}/3$*. In each location, the state of the system is governed by the stochastic differential equation*

$$\mathrm{d}\mathbf{x} = \mathrm{d}t + \mathrm{d}B_t, \tag{2.12}$$

The probability distribution $F(t,q,x)$ *of the state evolves by the Fokker-Planck equation*

$$\frac{\partial F(t,q,x)}{\partial t} = \frac{\partial F(t,q,x)}{\partial x} + \frac{1}{2}\frac{\partial^2 F(t,q,x)}{\partial x^2} \tag{2.13}$$

with the boundary conditions

$$\begin{aligned}
&\frac{\partial}{\partial x}F(t,q,0) = 0,\\
&\frac{\partial}{\partial x}F(t,q,1) = \frac{1}{2}\int_{[2,4]}\frac{\partial}{\partial t}F(t,q,x)\mathrm{d}x,\\
&\frac{\partial}{\partial x}F(t,q,2) = \int_{[1,2]}\frac{\partial}{\partial t}F(t,q,x)\mathrm{d}x,\\
&\frac{\partial}{\partial x}F(t,q,4) = 0.
\end{aligned} \tag{2.14}$$

Initially, the state of the system is uniformly distributed on $[0,1/2]$. *The goal is to check the following Metric Interval Temporal Logic formulas Section 2.3.2 in the system*

$$\varphi_1 = \mathsf{T}\mathcal{U}\left(y_2(t) > \frac{1}{4}\right), \tag{2.15}$$

$$\varphi_2 = \left(y_1(t) > \frac{1}{2}\right)\mathcal{U}\left(y_2(t) > \frac{1}{4}\right), \tag{2.16}$$

where

$$y_1(t) = \sum_{q\in\mathcal{Q}}\int_{\mathbb{R}^d} I_{[0,1]}F(t,q,x)\mathrm{d}x, \tag{2.17}$$

$$y_2(t) = \sum_{q\in\mathcal{Q}}\int_{\mathbb{R}^d} I_{[2,4]}F(t,q,x)\mathrm{d}x. \tag{2.18}$$

2.5.2 Discrete Time

Discretizing the time of the continuous-time stochastic hybrid system gives a discrete-time stochastic hybrid system with the initial distribution $F(0,q,x)$ and transition function $T(q',x',q,x)$, which is also a Markov kernel, satisfying

$$\sum_{q\in\mathcal{Q}}\int_{\mathbb{R}^d} T(q',x',q,x)\mathrm{d}x' = 1, \tag{2.19}$$

for any $(q,x) \in \mathbb{X}$. The observable y is defined in the same way as in the continuous-time case.

Definition 16. *A Markov kernel T is called strictly contractive by factor $\alpha \in$*

$(0,1)$ *if for any two distributions* $F(q,x), F'(q,x)$,

$$\|TF(q,x) - TF'(q,x)\|_{TV} \leq \alpha \|F(q,x) - F'(q,x)\|_{TV}, \tag{2.20}$$

For example, diffusive processes on compact state spaces are strictly contractive.

2.6 Differential Privacy

The concept of Differential Privacy (DP) is initially proposed to provide a measure of the chances of identifying a record from queries of statistical databases. More generally, it can be defined on a parametric probabilistic model, in which the probability distribution $f_X(x;\theta)$ of the observation $X \in \mathbb{R}^n$ depends on some *private data* $\theta \in \mathbb{R}^n$.

The requirement of differential privacy gives an upper bound on how much the probability distribution function $f_X(x;\theta)$ can change with the private data θ. Let $\|\cdot\|$ be a norm on the private data θ, two versions, non-metric and metric, of ϵ-Differential Privacy can be defined, in which the Metric version is stronger than the Non-Metric version.

Definition 17 (Differential Privacy)**.** *The parametric probabilistic model is ϵ-Differentially Private, if for any $\theta, \theta', x \in \mathbb{R}^n$ and set of possible observations $O \subseteq \mathbb{R}^n$, the probability distribution function satisfies*

$$\begin{aligned} &\textit{(Non-Metric)} && \int_O f_X(x;\theta)\mathrm{d}x \leq e^{\epsilon} \int_O f_X(x;\theta')\mathrm{d}x, \textit{ for } \|\theta - \theta'\| \leq 1, \\ &\textit{(Metric)} && \int_O f_X(x;\theta)\mathrm{d}x \leq e^{\epsilon\|\theta-\theta'\|} \int_O f_X(x;\theta')\mathrm{d}x. \end{aligned}$$

Given a deterministic query, a common way to make it ϵ-Differentially Private is to add Laplace noise. A Laplace noise v with parameter λ, written as $v \sim \mathrm{Lap}(\lambda)$, has a probability distribution function

$$f_{\text{Laplace}}(x) = \frac{1}{2\lambda}\exp(-\frac{|x|}{\lambda}). \tag{2.21}$$

The definition extends to n-dimensional random vectors by using the ℓ_1-norm, namely, $w \sim \text{Lap}(\lambda, n)$ if

$$f_{\text{Laplace}}(x) = \left(\frac{1}{2\lambda}\right)^n \exp(-\frac{\|x\|_1}{\lambda}). \tag{2.22}$$

Note that the components of the Laplace random vector are independent.

Finally, for the parametric probabilistic model, the *(Shannon) entropy* of the randomized observation is given by

$$H(X) = -\int_{\mathbb{R}^n} f_X(x)\ln(f_X(x))\mathrm{d}x. \tag{2.23}$$

Part I

Statistical Verification

CHAPTER 3

MODEL REDUCTION OF STOCHASTIC HYBRID SYSTEMS

In this chapter, the model reduction technique is demonstrated on the Discrete-Time and Continuous-Time Stochastic Hybrid Systems to reduce them to Discrete-Time Markov Chains and Continuous-Time Markov Chains, respectively.

3.1 Discrete Time

3.1.1 Reducing the Dynamics

A Discrete-Time Stochastic Hybrid Systems can be reduced to a Discrete-Time Markov Chain using the set-oriented methods [90]. Let $S = \{s_1, s_2, \ldots, s_n\}$ be a partition of the continuous state space $\mathbb{X}$, and P, R be the corresponding projection and injection operators as given by (3.16)-(3.18). As shown in Fig. 3.1 and Theorem 1, they induce a projection from the Markov kernel $T : m(\mathbb{X}) \to m(\mathbb{X})$ to a Markov kernel $T_r : m(S) \to m(S)$ by

$$T_r = PTR. \tag{3.1}$$

For multiple steps, the diagram for projection is shown by the non-commutative diagram in Fig. 3.2.

$$\begin{tikzcd}
F(t,q,x) \arrow[r, "T"'] & F(t+1,q,x) \arrow[d, "P"] \\
p(t) \arrow[u, "R"] \arrow[r, "T_r"'] & p(t+1)
\end{tikzcd}$$

Figure 3.1: Diagram for single-step reduction

$$\begin{tikzcd}
F(0,q,x) \arrow[r, "T"'] & F(1,q,x) \arrow[r, dotted] & F(t-1,q,x) \arrow[r, "T"'] & F(t,q,x) \arrow[d, "P"] \\
p(0) \arrow[u, "R"] \arrow[r, dashed, "T_r"'] & p(1) \arrow[r, dotted] & p(t-1) \arrow[r, dashed, "T_r"'] & p(t)
\end{tikzcd}$$

Figure 3.2: Diagram for multiple-step reduction

Theorem 1. *Let $S = \{s_1, \ldots, s_n\}$ be a measurable partition of the state space $\mathbb{X}$. Then the discrete-time stochastic hybrid system reduces to a CTMC (T_r, p_0) by*

$$\begin{aligned}
p_0(i) &= \int_{s_i} F(0,q,x)\mathrm{d}x, \\
T_r(i,j) &= \int_{s_i}\int_{s_j} T(q',x',q,x)\mathrm{d}x'\mathrm{d}x.
\end{aligned} \tag{3.2}$$

3.1.2 Reduced iLTL

An observable on the discrete stochastic hybrid system can be reduced approximately to an observable on the discrete-time Markov chain by (3.21). At a time t, discrepancy between $y(t)$ and $y'(t)$ is given by (1).

Lemma 1. *For any $F(q,x) \in m(\mathbb{X})$ and projection operator P, the following statements hold:*

$$\begin{aligned}
y(0) > b + \delta_P(F)\|F\|_\infty &\Longrightarrow y'(0) > b, \\
y'(0) > b + \delta_P(F)\|F\|_\infty &\Longrightarrow y(0) > b, \\
y(0) < b - \delta_P(F)\|F\|_\infty &\Longrightarrow y'(0) < b, \\
y'(0) < b - \delta_P(F)\|F\|_\infty &\Longrightarrow y(0) < b,
\end{aligned}$$

where

$$\delta_P(F) = \|F(0,q,x) - PF(0,q,x)\|_{TV}, \tag{3.3}$$

is the error of projection operator P in total variance.

Therefore, the iLTL formulas associated with the discrete-time stochastic hybrid system are reduced to iLTL formulas associated with the Markov Chains

(T_r, p_0) by replacing the integration with the summation. The *reduced iLTL* is exactly the form proposed in [91].

3.1.3 Reduction Error Estimation

First, the projection operator P is contractive.

Lemma 2. *Let $\mathbb{S} = \{s_1, \ldots, s_n\}$ be a measurable partition of $\mathbb{X}$ and P be the projection operator associated with $\mathbb{S}$. For any $F(q,x), F'(q,x) \in m(\mathbb{X})$,*

$$\|PF(q,x) - PF'(q,x)\|_{TV} \leq \|F(q,x) - F'(q,x)\|_{TV}. \tag{3.4}$$

As shown in the non-commutative diagram in Fig. 3.2, the discrepancy for t steps is

$$\begin{aligned}\Delta_t &= \|PT^{(t)}F(0,q,x) - T_r^{(t)}PF(0,q,x)\|_{\text{TV}} \\ &= \|PT^{(t)}F(0,q,x) - P(TRP)^{(t)}F(0,q,x)\|_{\text{TV}}.\end{aligned} \tag{3.5}$$

The error bound of t-step projection is given by the following theorem.

Theorem 2. *Given a discrete-time stochastic hybrid system and a projection operator P, the t-step ($t \geq 1$) error of projection*

$$\Delta_t \leq \sum_{i=0}^{t-1} \delta_P((TRP)^{(i)}F(0,q,x)), \tag{3.6}$$

where δ_P is given in (3.3).

Proof. For $t = 1$,

$$\begin{aligned}\Delta_1 &= \|PTF(0,q,x) - P(TRP)F(0,q,x)\|_{\text{TV}} \\ &\leq \|TF(0,q,x) - TRPF(0,q,x)\|_{\text{TV}} \\ &\leq \|F(0,q,x) - RPF(0,q,x)\|_{\text{TV}} \\ &= \delta_P(F(0,q,x)).\end{aligned} \tag{3.7}$$

For $t > 1$, with F denoting $F(0, q, x)$,

$$\begin{aligned}\Delta_t =&\|PT^{(t)}F - P(TRP)^{(t)}F\|_{\text{TV}} \\ \le&\|T^{(t)}F - (TRP)^{(t)}F\|_{\text{TV}} \\ \le&\|T^{(t)}F - T^{(t-1)}(TRP)F\|_{\text{TV}} \\ &+ \|T^{(t-1)}(TRP)F - T^{(t-2)}(TRP)^{(2)}F\|_{\text{TV}} \\ &+ \ldots + \|T(TRP)^{(t-1)}F - (TRP)^{(t)}F\|_{\text{TV}} \\ \le&\sum_{i=0}^{t-1} \delta_P((TRP)^{(i)}F).\end{aligned} \tag{3.8}$$

□

When T is strictly contractive, there exists a uniform error bound.

Theorem 3. *Given a discrete-time stochastic hybrid system, a projection operator P and the corresponding injection R, if the Markov kernel T is strictly contractive by factor $\alpha \in (0, 1)$, then the t-step ($t \geq 1$) error of projection*

$$\Delta_t \le \frac{\delta_P}{1-\alpha}, \tag{3.9}$$

where

$$\delta_P = \sup_{i\in\mathbb{N}} \delta_P((TRP)^{(i)}F(0, q, x)). \tag{3.10}$$

Proof. For $t = 1$, clearly $\Delta_t = \delta_P$. For $t \geq 2$, by (3.8) and with F denoting $F(0, q, x)$,

$$\begin{aligned}\Delta_t \le&\|T^{(t)}F - T^{(t-1)}(TRP)F\|_{\text{TV}} \\ &+ \|T^{(t-1)}(TRP)F - T^{(t-2)}(TRP)^{(2)}F\|_{\text{TV}} \\ &+ \ldots + \|T(TRP)^{(t-1)}F - (TRP)^{(t)}F\|_{\text{TV}} \\ \le&(1 + \alpha + \ldots + \alpha^t)\delta_P \\ \le&\frac{\delta_P}{1-\alpha}.\end{aligned} \tag{3.11}$$

□

By combining Lemma 1 and Theorem 3, the following theorem can be derived on the relationship between linear inequalities on the original Markov process and linear inequalities on the reduced Markov process.

Theorem 4. *Given a measurable partition $\mathbb{S} = \{s_1, \dots, s_n\}$ and the corresponding projection operator P, a discrete-time stochastic hybrid system and its reduction (T_r, p_0) satisfies the equations:*

$$y > b + \frac{\delta_P \|F\|_\infty}{1-\alpha} \Longrightarrow y' > b, \tag{3.12}$$

$$y' > b + \frac{\delta_P \|F\|_\infty}{1-\alpha} \Longrightarrow y > b, \tag{3.13}$$

$$y < b - \frac{\delta_P \|F\|_\infty}{1-\alpha} \Longrightarrow y' < b, \tag{3.14}$$

$$y' < b - \frac{\delta_P \|F\|_\infty}{1-\alpha} \Longrightarrow y < b, \tag{3.15}$$

where δ_p is given by (3.10) *respectively.*

3.2 Continuous Time

3.2.1 Reducing the Dynamics

To implement the Mori-Zwanzig model reduction method [19] for continuous-time stochastic systems, the continuous state space is divided into finitely many partitions $\mathbb{S} = \{s_1, \dots, s_n\}$, and treat each of them as a discrete state. It is assumed that for each s_i, there exists $q \in \mathcal{Q}$ such that $s_i \subseteq \{q\} \times \mathbb{A}_q$, and denote its measure by $\mu(s_i)$. Let $m(\mathbb{X})$ and $m(\mathbb{S})$ be set of probability distribution functions on $\mathbb{X}$ and $\mathbb{S}$, respectively. Define a projection $P : m(\mathbb{X}) \to m(\mathbb{S})$ and an injection $R : m(\mathbb{S}) \to m(\mathbb{X})$ between $m(\mathbb{X})$ and $m(\mathbb{S})$ by

$$p_j = (PF(q,x))_j = \int_{s_j} F(q,x)\mathrm{d}x, \tag{3.16}$$

where p_j is the jth element of p, and

$$Rp = \sum_{j=1}^{n} p_j \mathbb{I}_{s_j}, \tag{3.17}$$

where $\mathbb{I}_{s_j}$ is the uniform distribution on s_j:

$$\mathbb{I}_{s_j}(x) = \begin{cases} \frac{1}{\mu(s_j)}, & \text{if } x \in s_j \\ 0, & \text{otherwise.} \end{cases} \tag{3.18}$$

Here the projection P and the injection R are defined for probability distributions. But they extend naturally to L_1 functions on $\mathbb{X}$ and $\mathbb{S}$ respectively. The projection P is the left inverse of the injection R but not *vice versa*, namely $PR = I$ but $RP \neq I$.

This projection P and injection R can reduce the Fokker-Planck operator to a transition rate matrix on $\mathbb{S}$, and hence reduce the continuous-time stochastic hybrid system into a continuous-time Markov chain.

Theorem 5. *Let $\mathbb{S} = \{s_1, s_2, \ldots, s_n\}$ be a partition of the continuous state space $\mathbb{X}$ and P, R be the corresponding projection and injection defined in* (3.16)-(3.18). *The Fokker-Planck operator given in* (2.9) *reduces to the transition rate matrix A of a continuous-time Markov chain on $\mathbb{S}$ by*

$$A = PLR \tag{3.19}$$

where the transition rate from state s_i to s_j at time t is given by

$$\begin{aligned} A_{ij} = &\int_{\partial s_i \cap \partial s_j} f(q, x) \mathrm{d}x \\ &+ \frac{1}{\mu(s_i)} \int_{s_i} r(q, x) \mathbf{I}_{h(q,x) \in s_j} \mathrm{d}x \end{aligned} \tag{3.20}$$

for $a, b = 1, \ldots, n$, where $\mathbf{I}_{h(q,x)\in s_j} = 1$ when $h(q, x) \in s_j$, and 0 otherwise.

Roughly speaking, the transition rate between two partitions in the same location is the flux of $f(q, x)$ across the boundary and the transition rate between two different locations is the flux of $r(q, x)$.

3.2.2 Reducing MITL Formulas

The observables on the continuous-time stochastic hybrid system reduce to the corresponding continuous-time Markov chain using the projection P. Let y be an observable on the continuous-time stochastic hybrid system with weight function

$\gamma(q,x)$. To facilitate further discussion, it is assumed that $\gamma(q,x)$ is invariant under the projection P, *i.e.*, $\gamma(q,x) = P\gamma(q,x)$. Define a corresponding observable y' on the continuous-time Markov chain that derives from the model reduction procedure by

$$\begin{aligned} y'(0) &= \sum_{q\in\mathcal{Q}} \int_{\mathbb{R}^d} \gamma(q,x) PF(0,q,x)\mathrm{d}x \\ &= \sum_{i=1}^{n} \left(\int_{s_i} \gamma(q,x)\mathrm{d}x\right)\left(\int_{s_i} F(0,q,x)\mathrm{d}x\right) \\ &= \sum_{i=1}^{n} r_i p(i) = y'(0). \end{aligned} \tag{3.21}$$

In the rest of this section, denote the corresponding observable on the CTMC by y' for any observable y on the continuous-time stochastic hybrid system.

For a given observable y with weight function $\gamma(q,x)$, the error of the projection P with respect to the observable y is defined by the maximal possible difference between y and y',

$$\Delta_y = \Big|\sum_{q\in\mathcal{Q}} \int_{\mathbb{R}^d} \gamma(q,x)(F(0,q,x) - RPF(0,q,x))\mathrm{d}x\Big|. \tag{3.22}$$

Remark 1. *When refining the partition of $\mathbb{X}$, $RP \to I$ in the weak operator topology, thus $\Delta_y \to 0$ for any given y.*

By the definition of Δ_y, at the initial time, the atomic propositions on the continuous-time stochastic hybrid system and the CTMC have the relations

$$y(0) > c \Longrightarrow y'(0) > c - \Delta_y, \tag{3.23}$$

$$y(0) < c \Longrightarrow y'(0) < c + \Delta_y, \tag{3.24}$$

and similarly,

$$y'(0) > c + \Delta_y \Longrightarrow y(0) > c, \tag{3.25}$$

$$y'(0) < c - \Delta_y \Longrightarrow y(0) < c. \tag{3.26}$$

To derive the relations of the observables between the continuous-time stochastic hybrid system and the CTMC at any time, define the reduction error of the

$$\begin{tikzcd}
F(0,q,x) \arrow[r, "e^{Lt}"'] \arrow[d, dashed, "P"'] & F(t,q,x) \\
p(0) \arrow[r, "e^{At}"'] & p(t) \arrow[u, dashed, "R"']
\end{tikzcd}$$

Figure 3.3: Diagram for reduction error.

observable y at time t due to the model reduction process by

$$\begin{aligned}\Theta_y(t) &= |y(t) - y'(t)| \\ &= \Big|\sum_{q\in\mathcal{Q}} \int_{\mathbb{R}^d} \gamma(q,x)(e^{Lt} - Re^{At}P)F(0,q,x)\mathrm{d}x\Big|,\end{aligned} \tag{3.27}$$

where $F(0,q,x)$ is an initial distribution of the continuous-time stochastic hybrid system and $y'(t)$ is the corresponding observable of $y(t)$ on the CTMC. This reduction error is illustrated in Fig. 3.3. Note that the diagram is not commutative; actually the difference between going along the two paths is related to the reduction error.

In general, the reduction error $\Theta(t)$ may not be bounded as $t \to \infty$. To find a sufficient condition for boundedness, define the reduction error of the Fokker-Planck operator L by

$$\delta(t,q,x) = (L - RPL)e^{tRPL}F(0,q,x). \tag{3.28}$$

Accordingly, define the integration of $\delta(t,q,x)$ with respect to the weight function $\gamma(q,x)$ by

$$\Lambda_y = \sup_{t\geq 0} \Big|\sum_{q\in\mathcal{Q}} \int_{\mathbb{R}^d} \gamma(q,x)(L - RPL)e^{tRPL}F(0,q,x)\mathrm{d}x\Big|, \tag{3.29}$$

which captures the maximal change of the time derivative of observable y.

A sufficient condition to find a uniform bound over time is that the reduction error of the Fokker-Planck operator $\delta(f(q,x))$ converges exponentially in time for any $f(q,x) \in m(\mathbb{X})$.

Definition 18. *For $\alpha > 0$, $\beta \geq 1$ and a given observable y, the continuous-time stochastic hybrid system is α-contractive with respect to y, if for any initial* *distribution function $F(0, q, x)$ on the state space,*

$$\begin{aligned}&\Big|\sum_{q\in\mathcal{Q}}\int_{\mathbb{R}^d}\gamma(q,x)e^{tL}\delta(t,q,x)\mathrm{d}x\Big|\\&\qquad\leq \beta e^{-\alpha t}\Big|\sum_{q\in\mathcal{Q}}\int_{\mathbb{R}^d}\gamma(q,x)\delta(t,q,x)\mathrm{d}x\Big|.\end{aligned}\tag{3.30}$$

where $\delta(t, q, x)$ is given by (3.28).

This contractivity condition, though it seems restrictive, is valid for a relatively wide range of systems including asymptotically stable systems. It is a commonly-used sufficient condition to guarantee the existence and uniqueness of an invariant measure for general dynamical systems, and the contractivity factor α is usually derived case-by-case.

Theorem 6. *If the continuous-time stochastic hybrid system is α-contractive, then for any $t \geq 0$, the reduction error $\Theta_y(t)$ for an observable y satisfies*

$$\Theta_y(t) \leq \frac{\beta\Lambda_y}{\alpha} + \Delta_y.\tag{3.31}$$

Proof. By Dyson's formula, the exponential of L is decomposed by

$$e^{tL} = e^{tRPL} + \int_{[0,t]} e^{(t-\tau)L}(L - RPL)e^{\tau RPL}\mathrm{d}\tau.\tag{3.32}$$

This formula, sometimes referred to as Duhamel's principle, can be verified by taking time derivatives on both sides. Substituting (3.32) into (3.27) gives

$$\begin{aligned}\Theta_y(t) \leq &\Big|\sum_{q\in\mathcal{Q}}\int_{\mathbb{R}^d}\gamma(q,x)(e^{tRPL} - Re^{tA}P)F(0,q,x)\mathrm{d}x\Big|\\+&\Big|\sum_{q\in\mathcal{Q}}\int_{\mathbb{R}^d\times[0,t]}\gamma(q,x)e^{(t-\tau)L}(L-RPL)e^{\tau RPL}F(0,q,x)\mathrm{d}\tau\mathrm{d}x\Big|\end{aligned}\tag{3.33}$$

Since the projection P and the injection R preserve the L_1 norm, RPL is also a Fokker-Planck operator. Noting $Re^{tA}PF(0, q, x) = e^{tRPL}PF(0, q, x)$, by (3.22), the first term on the right hand side of (3.33) is less than Δ_y.

By (3.29)-(3.30), the second term on the right hand side of (3.33) satisfies

$$\begin{aligned}\Theta_y(t) &\leq \Delta_y + \Big|\sum_{q\in\mathcal{Q}}\int_{\mathbb{R}^d}\int_{[0,t]}\gamma(q,x)e^{(t-\tau)L}\delta(\tau,q,x)\mathrm{d}\tau\mathrm{d}x\Big| \\ &\leq \Delta_y + \Big|\sum_{q\in\mathcal{Q}}\int_{\mathbb{R}^d}\int_{[0,t]}\beta e^{-\alpha(t-\tau)}\gamma(q,x)\delta(\tau,q,x)\mathrm{d}\tau\mathrm{d}x\Big| \\ &\leq \frac{\beta\Lambda_y}{\alpha} + \Delta_y.\end{aligned} \tag{3.34}$$

□

Theorem 6 implies the following relations between the atomic propositions on the continuous-time stochastic hybrid system and the CTMC.

Theorem 7. *If the continuous-time stochastic hybrid system is α-contractive, then*

$$y(t) > c \Longrightarrow y'(t) > c - \Big(\frac{\beta\Lambda_y}{\alpha} + \Delta_y\Big), \tag{3.35}$$

$$y(t) < c \Longrightarrow y'(t) < c + \Big(\frac{\beta\Lambda_y}{\alpha} + \Delta_y\Big), \tag{3.36}$$

and similarly,

$$y'(t) > c + \Big(\frac{\beta\Lambda_y}{\alpha} + \Delta_y\Big) \Longrightarrow y(t) > c, \tag{3.37}$$

$$y'(t) < c - \Big(\frac{\beta\Lambda_y}{\alpha} + \Delta_y\Big) \Longrightarrow y(t) < c. \tag{3.38}$$

The above theorem gives the following result.

Theorem 8. *Given a MITL formula φ on the continuous-time stochastic hybrid system that is α-contractive, it can be strengthened to ψ by replacing the atomic propositions according to* (3.37)-(3.38)*. If ψ is true on the corresponding CTMC, then φ is true on the continuous-time stochastic hybrid system.*

Example 2. *First, the invariant distribution of this process is $F_{\text{inv}} = \mathbb{I}_{\mathbb{X}}/3$. Following Example 1, $\mathbb{X}$ is partitioned into intervals of length $1/N$. By the above model reduction procedure it reduces to a CTMC with transition matrix M given by*

$$M_{ij} = \frac{\delta_{ij}}{4} + \frac{1}{4N} \tag{3.39}$$

where $i, j \in [3N]$. The invariant distribution F_{inv} remains unchanged, and the MITL formula to check is

$$\varphi_1' = \mathsf{T}\mathcal{U}\left(y_2'(t) > \frac{1}{4} + \delta(N)\right) \tag{3.40}$$

$$\varphi_2' = \left(y_1'(t) > \frac{1}{2} + \delta(N)\right)\mathcal{U}\left(y_2'(t) > \frac{1}{4} + \delta(N)\right) \tag{3.41}$$

where $\delta(N)$ is the model reduction error and

$$y_1'(t) = \sum_{i=1}^{N} p_i(t), \tag{3.42}$$

$$y_2'(t) = \sum_{i=2N+1}^{3N} p_i(t). \tag{3.43}$$

When $N = 30$, it can be computed that $\delta(N) \leq 0.02$ from (3.16) *and* (3.29).

CHAPTER 4

STATISTICAL VERIFICATION OF ILTL AND MITL ON MARKOV CHAINS

In this chapter, statistical verification algorithms are proposed for checking Linear Inequality LTL on Discrete-Time Markov Chains and Metric Interval Temporal Logic on Continuous-Time Markov Chains.

4.1 Linear Inequality LTL

Denote the atomic proposition $y = \sum_{i=1}^{n} r_i p_i = r \cdot p$ by a pair (r, b). For an iLTL formula φ and a discrete-time Markov chain generating a sequence of distributions $w = p_0 p_1 p_2 \ldots$, define $u = u_0 u_1 u_2 \ldots$ where $u_t = \{(r, b) \in \mathsf{AP}_\varphi \mid r \cdot p_t > b\}$ is the set of atomic propositions that are true at time t. Then $w \models \varphi$, iff $u \models \varphi$. As mentioned in Section 2.1.3, φ can be transformed into a Büchi automaton B_φ such that $[\![\varphi]\!] = \texttt{Lang}(B_\varphi)$, meaning B_φ accepts exactly those sequences that satisfy φ. This suggests the following algorithm to check if a Markov chain satisfies an iLTL formula φ. Let w be the (unique) sequence of distributions generated by that Markov chain.

1. Construct the sequence $u \in (2^{\mathsf{AP}})^{\boldsymbol{\omega}}$ of atomic propositions that are true at each step of w.

2. Check $u \in \texttt{Lang}(B_\varphi)$ and return the result as the algorithm output. It will be the right answer since $(T_r, p_0) \models \varphi$ iff $w \models \varphi$ iff $u \models \varphi$ iff $u \in \texttt{Lang}(B_\varphi)$.

The details on the two steps are given below.

Constructing the labels for distributions To construct the set of labels u_t corresponding to the distribution $w_t = p_t$, the simplest algorithm would compute $p_t = T_r^{(t)} p_0$ first and then for every atomic proposition $(r, b) \in \mathsf{AP}_\varphi$, check whether or not $r \cdot p_t > b$ is true. However, this would be expensive for Markov chains with a large number of states. Instead, these labels are computed statistically. First, draw samples according to distribution p_t by simulating the Markov chain for t steps. Assuming the elements of each vector r in atomic propositions are from $\{0, 1\}$. In this case, p_t satisfies (r, b) iff the probability of drawing a state s (according to p_t) such that $r_s = 1$ is strictly greater than b. This can be statistically checked by drawing samples from p_t and using either Chernoff bounds, or the Sequential Probability Ratio Test [92] (see [32] and [31]). Such a statistical test usually takes as parameters an indifference parameter $\delta_2 > 0$, and error bounds $\alpha_2, \gamma_2 > 0$. The output of this test is `yes`, `no`, or `unknown` with conditions:

$$\mathbb{P}[\texttt{res} = \texttt{no} \quad \mid r \cdot p_t > b \quad] \leq \alpha_2, \tag{4.1a}$$

$$\mathbb{P}[\texttt{res} = \texttt{yes} \quad \mid r \cdot p_t \not> b \quad] \leq \alpha_2, \tag{4.1b}$$

$$\mathbb{P}[\texttt{res} = \texttt{unknown} \mid |r \cdot p_t - b| > \delta_2] \leq \gamma_2. \tag{4.1c}$$

The parameters $\delta_2, \alpha_2, \gamma_2$ can be made arbitrarily small, though that will increase the number of samples needed. The general case when elements of r can be arbitrary real numbers requires one to estimate the mean of a random variable that is not necessarily Bernoulli. In such a situation, the Sequential Probability Ratio Test cannot be used, but a technique due to Chow and Robbins can be used [93].

Running B_φ on the labels The sequence of labels $u = u_0 u_1 \ldots$ can be constructed statistically, symbol by symbol. However, u is an infinite sequence, and in order to run B_φ on u, u needs to *ultimately periodic*, *i.e.*, there must be finite sequences u_1 and u_2 such that $u = u_1 u_2^{\boldsymbol{\omega}}$. It is assumed that the mapping defined by the reduced model is contracting and hence the sequence w converges to the invariant distribution p^{inv} that is known for some bounded uncertainty. This assumption is readily verified for large classes of important physical models, such as those with energy balance laws that are dissipative in aggregate. For such models, general arguments can be used to derive the prior condition, even in the presence of strong nonlinearities, discontinuous dynamics, or other complexities. Define $\eta = \max_{(r,b)\in\mathsf{AP}} \|r\|_1$ to be the size of largest vector in atomic propositions and without loss of generality, assume it is positive. It is assumed $\delta' > 0$ and p^* (an estimate of p^{inv}) are given such that $\forall (r, b) \in \mathsf{AP}, |r \cdot p^{\text{inv}} - b| > \delta'$ and $\|p^{\text{inv}} - p^*\|_1 < \frac{\delta'}{3\eta}$ are both true. Since w converges to p^{inv}, for any atomic proposition (r, b), $|r \cdot p^{\text{inv}} - b| > \delta'$, for large enough t. p_t and p^{inv} will satisfy exactly the same propositions and so u is ultimately periodic. Also, note that if p^{inv} is known precisely then it can be checked if such δ' exists; otherwise, the supremum of all such δ' can be found.

1. Using $\mathcal{A}$, with parameters $\frac{1}{2}\min\{\alpha, \gamma\}$ and $\frac{\delta'}{3\eta}$ find m such that p_m is within distance $\frac{\delta'}{3\eta}$ of p^* and hence within distance $\frac{2\delta'}{3\eta}$ of p^{inv} (see function `NumberOfSamplingSteps` in Algorithm 1). Let $(r, b) \in \mathsf{AP}_\varphi$ be an arbitrary atomic proposition, and let p be any distribution for which $\|p - p^{\text{inv}}\| \leq \frac{2\delta'}{3\eta}$. Clearly this includes p^* which is known exactly, and any distribution $p_{m'}$ for $m' \geq m$.

$$|r \cdot p - r \cdot p^{\text{inv}}| \leq \|r\|_1 \times \frac{2\delta'}{3\eta} \leq \frac{2\delta'}{3} \tag{4.2}$$

   From (4.2), if $r \cdot p^{\text{inv}} - b > \delta'$ then $r \cdot p - b > \delta' - \frac{2}{3}\delta' > 0$, and if $r \cdot p^{\text{inv}} - b < -\delta'$ then $r \cdot p - b < -\delta' + \frac{2}{3}\delta' < 0$. Therefore, assuming a proper m, truth values of atomic propositions do not change in $p_m p_{m+1} \ldots$.

2. For each $t < m$ and atomic proposition $(b, r) \in \mathsf{AP}$, use $\mathcal{A}$ with parameters $\delta_2 = \delta$, $\alpha_2 = \frac{\alpha}{2m|\mathsf{AP}|}$, and $\gamma_2 = \frac{\gamma}{2m|\mathsf{AP}|}$, to determine if $r \cdot p_t > b$ (see function `LabelFiniteNumberOfSteps` in Algorithm 1).

3. For every atomic proposition $(b, r) \in \mathsf{AP}$, verify if $r \cdot p^* > b$ is true or not (see function `AddLabelsOfInvariantDistribution` in Algorithm 1). Note that there will be no error at this step, since p^* is known precisely. Also, after this step, asg is defined for all $t \in \mathbb{N}$ and $q \in \mathsf{AP}$. Whenever $asg(t, q) =$ `unknown`, consider both possibilities. With this consideration, function asg induces a set of infinite sequences on alphabet 2^{AP}. Since after m, asg is a constant function, this set can be easily represented by a Büchi automaton $[\![asg]\!]$.

4. Verify if `Lang`(B_φ) and its complement has any intersection with `Lang`$([\![asg]\!])$. If both intersections are non-empty, it means different choices for `unknown` slots (*i.e.*, `yes` or `no`) can lead to both satisfying φ and not satisfying it. Therefore, the algorithm returns `unknown`. Otherwise, if `Lang`(B_φ) has no intersection with `Lang`$([\![asg]\!])$, the resulting infinite sequence can never satisfy φ, hence the algorithm give `no`. The only remaining case is when `Lang`$(B_{\neg\varphi})$ has no intersection with `Lang`$([\![asg]\!])$, and using a similar argument, the right answer will be `yes`. Note that, `Lang`$(B_{\neg\varphi})$ is exactly complement of the set `Lang`(B_φ), a fact which is used multiple times.

The algorithm $\mathcal{A}$ outlined above provides the following guarantees:

$$\mathbb{P}[\mathcal{A}((T_r, p_0), \varphi, \alpha, \gamma) = \texttt{no} | (T_r, p_0) \models \varphi] \leq \alpha \tag{4.3a}$$

$$\mathbb{P}[\mathcal{A}((T_r, p_0), \varphi, \alpha, \gamma) = \texttt{yes} | (T_r, p_0) \not\models \varphi] \leq \alpha \tag{4.3b}$$

$$\begin{aligned} &\mathbb{P}\left[\mathcal{A}((T_r, p_0), \varphi, \alpha, \gamma) = \texttt{unknown} |\right. \\ &\quad \left.\nexists(b, r) \in \mathsf{AP}, |r \cdot p_t - b| \leq \delta\right] \leq \gamma \end{aligned} \tag{4.3c}$$

The first two inequalities state that probability of having false positive or negative is at most α. The last inequality states that if in all steps that have distributions far enough from the invariant distribution, the actual probability of no atomic proposition (r, b) in φ is too close to b then the probability of returning unknown is at most γ.

The error analysis of the algorithm can be carried out as follows. When the algorithm returns no (yes) while the correct answer is yes (no), it means that the algorithm made at least one mistake. The probability of finding wrong m is at most $\frac{\alpha}{2}$. Assuming m is computed correctly, the probability of having a step t and an atomic formula $q \in \mathsf{AP}$ such that truth value of q at step t is computed incorrectly is at most $(m|\mathsf{AP}|)\frac{\alpha}{2m|AP|} = \frac{\alpha}{2}$ (here unknown is considered a correct answer, because it did not effect the output of the algorithm). Therefore, the total error is at most $\alpha = \frac{\alpha}{2} + \frac{\alpha}{2}$.

Similarly, if the algorithm returns unknown while for any step that is far enough from the invariant distribution, the actual probability of no atomic proposition is too close to the threshold of that proposition, it means either the algorithm found m incorrectly, or it found unknown for at least one step and one atomic proposition incorrectly. But the probability of making each of these mistakes is at most $\frac{\gamma}{2}$. Thus the probability of incorrectly returning unknown is at most γ.

$\mathcal{A}$ takes δ as one of its parameters. The problem with δ is that one may not know in advance the correct value for δ. Large values cause the algorithm to return unknown, and small values make the algorithm slow. In order to solve this problem one can start with a a large value for δ and decrease it when the algorithm returns unknown for that δ.

4.2 Metric Interval Temporal Logic

Given a CTMC C and a MITL formula φ with atomic propositions AP_φ, a timed automaton $T_{C,\mathsf{AP}_\varphi}$ can be constructed by sampling, whose reachable locations at time t are labeled by the atomic propositions in φ that are true on C. By $[\![C, \mathsf{AP}_\varphi]\!]$ denote the singleton set containing the unique signal induced by C and φ. For simplicity, consider constructing $T_{C,\{P\}}$ for a single atomic formula $P : y = \sum_{i=1}^{n} r_i p_i > c$, denoted by a pair (r, c). Let $f(t)$ be the set of atomic formulas that p satisfies at time t, *i.e.*, $(p, c) \in f(t)$ iff $p(t) > c$. Also, let $T_{C,\{P\}}(t)$ be the set of reachable locations of $T_{C,\{P\}}$ at time t.

Lemma 3. *[94] For any $\alpha, \delta > 0$, and two discrete distributions p and p', there is a test $\mathcal{A}(p, p', \alpha, \delta)$ which runs in time $O\big(n^{2/3}(2\delta)^{-8/3}\log(n/\alpha)\big)$ such that if $\|p - p'\| \leq \max\Big(\frac{\delta^{4/3}}{2^{14/3}\sqrt[3]{n}}, \frac{\delta}{4\sqrt{n}}\Big)$ then the test accepts with probability at least $1 - \alpha$, and if $\|p - p'\| > \delta$ then the test rejects with probability at least $1 - \alpha$.*

It is assumed an estimation p^* of the invariant distribution p^{inv} are given such that $\forall(p, c) \in \mathsf{AP}, |p^{\mathrm{inv}} - c| > \delta'$ and $|p^{\mathrm{inv}} - p^*| < \frac{\delta'}{3}$ for some $\delta' > 0$. Since $p(t)$ converges to p^{inv} due to contractivity, $|p(t) - c| > \delta'$, for large enough t. Using Lemma 3, the truncation time T can be found such that for $t > T$, $|p(t) - p^*| < \frac{\delta'}{3}$, namely $|p(t) - p^{\mathrm{inv}}| < \frac{2\delta'}{3}$. Therefore, if $p^{\mathrm{inv}} - c > \delta$ then $p^* - c > 0$. Similarly, if $p^{\mathrm{inv}} - c < -\delta$ then $p^* - c < 0$. Note that exactly one of $p^{\mathrm{inv}} - c > \delta$ and $p^{\mathrm{inv}} - c < -\delta$ is true. Furthermore, $p^* - c > 0$ and $p^* - c < 0$ cannot be both true. Therefore, $p^* - c > 0$ implies $p^{\mathrm{inv}} - c > \delta$, $p^* - c < 0$ implies $p^{\mathrm{inv}} - c < -\delta$, and $p^* - c$ is never zero.

For any $\delta_1 > 0$, let $\Delta = \frac{\delta_1}{3\max\{|\dot{p}_i(t)| | t \in [0,T]\}}$. Then, for any $t \in [0, T]$ and $t' \in [t - \Delta, t + \Delta] \cap [0, T]$,

1. if $p_i(t) - c > \frac{\delta_1}{3}$, then $p_i(t') - c > 0$,
2. if $p_i(t) - c < -\frac{\delta_1}{3}$, then $p_i(t') - c < 0$,
3. if $|p_i(t) - c| \leq \frac{2\delta_1}{3}$, then $|p_i(t') - c| \leq \delta_1$.

The time interval $[0, T)$ is partitioned into at least $\lfloor \frac{T}{2\Delta} \rfloor + 1$ intervals, each of size smaller than 2Δ. Let $[t_1, t_2]$ be one of these intervals and run $\mathcal{A}$, for $t = \frac{1}{2}(t_1 + t_2)$

to derive

$$\begin{aligned}\texttt{res}_1 &= \mathcal{A}^{\delta_1/3}\left(p_i(t), c + \frac{\delta_1}{3}, \alpha', \gamma'\right),\\ \texttt{res}_2 &= \mathcal{A}^{\delta_1/3}\left(p_i(t), c - \frac{\delta_1}{3}, \alpha', \gamma'\right),\end{aligned}$$

where $\mathcal{A}$ statistically check If $\texttt{res}_1 = \texttt{yes}$ then $\forall t' \in [t_1, t_2), (p_i(t') > c)$ holds with bounded error α'. Therefore, set $T_{C,\{P\}}(t) = \{P\}$. If $\texttt{res}_2 = \texttt{no}$ then for any time $t' \in [t_1, t_2]$, $p_i(t') < c$ holds with bounded error α'. Therefore, set $T_{C,\{P\}}(t) = \{\emptyset\}$. Otherwise, for any time t' in the interval, $|p_i(t') - c| \leq \delta_1$ with bounded error $\max(\alpha', \gamma')$. In this case, set

- $T_{C,\{P\}}(t) = \{q, q'\}$,
- $\mathsf{L}(q) = \{P\}$ and $\mathsf{L}(q') = \emptyset$,
- entry to q or q',
- switches between q and q' for arbitrary number of times, while their common invariant permits.

The result of the above procedure $\texttt{res} = \mathcal{A}^{\delta_1,\delta_2}(C, y_0, \varphi, \alpha, \beta)$ satisfies

$$\mathbb{P}[\texttt{res} = \texttt{no} \ \mid C \models \varphi] \leq \alpha \tag{4.4}$$

$$\mathbb{P}[\texttt{res} = \texttt{yes} \mid C \not\models \varphi] \leq \alpha \tag{4.5}$$

As for the $\texttt{unknown}$ output, let $B^{\delta_1}(y)$ be the δ_1-ball centered at y in the L_∞ norm. The algorithm guarantees that

$$\mathbb{P}[\texttt{res} = \texttt{unknown}] \leq \alpha + \beta \tag{4.6}$$

for all $y' \in B^{\delta_1}(y)$.

Definition 19. *For any $\epsilon > 0$ let $y + B_\epsilon$ be the set of observables achieved by slightly perturbing y. Let C_ϵ be any object with observables in the set $y + B_\epsilon$. The satisfaction relation of CTMC C and MITL formula φ is called* ϵ-robust*, if*

1. *For all y' induced by C_ϵ, $y' \models \varphi$, or*
2. *For all y' induced by C_ϵ, $y' \not\models \varphi$.*

The satisfaction relation is called robust, *if it is ϵ-robust for some $\epsilon > 0$.*

By definition 19, for any CTMC C and MITL formula φ, if C is robust on φ, iteratively reducing δ_1 in the algorithm guarantees that it will eventually return an answer which is not `unknown` while satisfying conditions (4.4) and (4.5).

Example 3. *Following Example 2, running Algorithm 2 on the CTMC shows that both φ_1' and φ_2' are true. This implies that the formulas φ_1 and φ_2 given are true on the Continuous-Time Stochastic Hybrid System in Example 1.*

Algorithm 1 Model checking Markov chains against iLTL formulas

```
Data: Markov chain $(T, p_0)$, estimation of invariant distribution $p^*$, iLTL formula
      $\varphi$, parameters $\alpha$, $\gamma$, $\delta$, and $\delta'$
Result: yes, no, or unknown
Function NumberOfSamplingSteps()
    $t \leftarrow 0$  $\eta \leftarrow \max_{(r,b) \in \mathsf{AP}} \|r\|_1$  while $\mathcal{A}\Big(p_t, p^*, \frac{1}{2}\min\{\alpha, \gamma\}, \frac{\delta'}{3\eta}\Big) =$ failed do
        $t \leftarrow t + 1$
    end
    return $t$
Function LabelFiniteNumberOfSteps($m \in \mathbb{N}$)
    forall $t \in \{0, 1, \ldots, m-1\}$, $(r, b) \in \mathsf{AP}$ do
        $asg(t, (r, b)) \leftarrow \mathcal{A}_2^{\delta}(p_t, r, b, \frac{\alpha}{2m|\mathsf{AP}|}, \frac{\gamma}{2m|\mathsf{AP}|})$
    end
    return $asg$
Function AddLabelsOfInvariantDistribution($m \in \mathbb{N}, asg \in \mathbb{N} \times \mathsf{AP} \rightarrow$
  $\{$yes, no, unknown$\}$)
    forall $t \in \{m, m+1, \ldots\}$, $(r, b) \in \mathsf{AP}$ do
        if $r \cdot p^* > b$ then
            $asg(t, (r, b)) \leftarrow$ yes
        else
            $asg(t, (r, b)) \leftarrow$ no
        end
    end
    return $asg$
Function ModelCheck
    $m \leftarrow$ NumberOfSamplingSteps()  $asg \leftarrow$
     LabelFiniteNumberOfSteps($m$)  $asg \leftarrow$
     AddLabelsOfInvariantDistribution($m, asg$)  $[\![asg]\!] \leftarrow$ the Büchi
     automaton that accepts exactly the set of infinite paths induced by $asg$  if
     Lang($B_\varphi$) $\cap$ Lang($[\![asg]\!]$) $\neq \emptyset \wedge$ Lang($B_{\neg\varphi}$) $\cap$ Lang($[\![asg]\!]$) $\neq \emptyset$ then
        return unknown
    else if Lang($B_\varphi$) $\cap\, asg = \emptyset$ then
        return no
    return yes
```

Algorithm 2 Truncating time horizon

```
Data: CTMC $(T, y_0)$, estimation of invariant distribution $y^*$, MITL formula $\varphi$,
      parameters $\alpha$, $\gamma$, $\delta$, and $\delta'$
Function DurationOfSimulation
    $t \leftarrow 0$
    $\eta \leftarrow \max_{(p,c)\in \mathsf{AP}} \|p\|_1$
    while $\mathcal{A}\Big(y_t, y^*, \frac{1}{2}\min\{\alpha,\gamma\}, \frac{\delta'}{3\eta}\Big) = \texttt{failed}$ do
        $t \leftarrow t + 1$
    end
    return $t$
```

Algorithm 3 Constructing the signal for atomic proposition P

```
$h \leftarrow \max\{|\dot{y}_i(t)| \mid t \in [0,T]\}$, $\Delta \leftarrow \frac{\delta_1}{3h}$, $n \leftarrow |\mathsf{AP}|\lceil\frac{T}{\Delta}\rceil$, $T_{C,\{P\}} \leftarrow$ an empty
automaton, $\mathsf{C} \leftarrow \{t\}$, $q_{\text{last}} \leftarrow \perp$
forall $i \leftarrow 0$ to $\lfloor\frac{T}{\Delta}\rfloor$ do
    $\alpha' \leftarrow \min(\frac{\alpha}{4n}, \frac{\beta}{2n})$, $\beta' \leftarrow \frac{\beta}{n}$
    $\texttt{res}_1 \leftarrow Algorithm_1^{\delta_1/3}\big(p_i\big((i+\frac{1}{2})\Delta\big), c + \frac{\delta_1}{3}, \alpha', \beta'\big)$
    $\texttt{res}_2 \leftarrow Algorithm_1^{\delta_1/3}\big(p_i\big((i+\frac{1}{2})\Delta\big), c - \frac{\delta_1}{3}, \alpha', \beta'\big)$
    add a new location $q$ to $\mathsf{S}$
    if $\texttt{res}_1 = \texttt{yes}$ then
        $\mathsf{L}(q) \leftarrow \{P\}$
    else if $\texttt{res}_2 = \texttt{no}$ then
        $\mathsf{L}(q) \leftarrow \emptyset$
    else
        $\mathsf{L}(q) \leftarrow \texttt{unknown}$
    $\mathsf{I}(q) \leftarrow 2i\Delta \leq t < 2(i+1)\Delta$
    if $q_{\text{last}} \neq \perp$ then
        $\mathtt{E} \leftarrow \mathtt{E} \cup \{(q_{\text{last}}, q, \emptyset)\}$
    else
        $\mathsf{s}_{\text{init}} \leftarrow \{q\}$
    $q_{\text{last}} = q$
end
add a new location $q$ to $\mathsf{S}$
$\mathsf{I}(q) \leftarrow \texttt{true}$, $\mathsf{S}_{\text{final}} \leftarrow \{q\}$
$\mathtt{E} \leftarrow \mathtt{E} \cup \{(q_{\text{last}}, q, \emptyset), (q, q, \emptyset)\}$
if $y^{\text{inv}} > c$ then
    $\mathsf{L}(q) \leftarrow \{P\}$
else
    $\mathsf{L}(q) \leftarrow \emptyset$
$T_{C,\{P\}} \leftarrow$ replace any unknown location in $\mathsf{S}$ with $q$ and $q'$ labeled $\{P\}$ and $\emptyset$.
Duplicate edges from/to $q$ and $q'$ accordingly
Add $(q, q', \emptyset)$ and $(q', q, \emptyset)$ to $\mathtt{E}$ for every split locations in the previous step.
return $T_{C,\{P\}}$
```

CHAPTER 5

STATISTICAL VERIFICATION OF PCTL USING STRATIFIED SAMPLES

Stratified sampling method is a popular method to generate negatively correlated samples that have lower variance. In this chapter, a statistical model checking algorithm is proposed for checking finite horizon Probabilistic Computation Tree Logic (PCTL) properties on Discrete-Time Markov Chains using stratified sampling. This algorithm significantly reduces the computational cost, compared to other algorithms using independent samples.

To ensure a lucid exposition of the main ideas, consider Probabilistic Computation Tree Logic formulas of the form $\mathbf{P}_{\sim p}\varphi$, where φ is a formula without probabilistic operator; in other words, φ's truth can be determined on a single path. PCTL formulas in general form with nested probabilistic operators can be handled in the standard manner using the approach proposed in [36, 37, 38]. The main result is a sequential probability ratio test that works when samples are drawn using stratified sampling, which helps reduce the total number of samples (number of strata × number of blocks of stratified samples) needed for a statistical model checker to be confident in its answer.

5.1 Markov Chains

Consider a discrete-time (homogeneous) Markov chain $\mathcal{M}$ of n numbered states with initial state $s \in [n]$ and transition probability matrix M, in which M_{ij} defines the transition probability from i to j. For any $j \in [n]$,

$$\sum_{i=1}^{n} M_{ij} = 1. \tag{5.1}$$

For a sample path $X = \{X(t)\}_{t\in\mathbb{N}} \subseteq [n]$ of the Markov chain,

$$X(t+1) = f(X(t), E(t)), \quad t \in \mathbb{N} \tag{5.2}$$

where $E(t) \sim \mathcal{U}_{[0,1)}$. Generally, a Discrete-Time Markov Chain $\mathcal{M}$ can be represented in (5.2) in multiple ways. In this work, the following representation is chosen

$$f(i,e) = \begin{cases} 1, & \text{if } 0 \leq e < M_{i1} \\ j, & \text{if } \sum_{k=1}^{j-1} M_{ij} \leq e < \sum_{k=1}^{j} M_{ij}. \end{cases} \tag{5.3}$$

5.2 Stratified Sampling

As shown in (5.2), the Markov chain is driven by the random seed $E(t)$ uniformly sampled from the interval $\mathcal{U}_{[0,1]}$. Therefore, there is a bijection between the space of sample paths of the Markov chain $\mathcal{M}$ of length T and $[0,1]^T$. The stratified sampling algorithm generates m sample paths simultaneously. At each time t, the interval $[0,1)$ can be partitioned into m sub-intervals, namely $[0,1] = [0, \frac{1}{m}] \cup \ldots \cup [\frac{m-1}{m}, 1)$. Thus, a sample can be drawn from each sub-interval. To avoid correlation between steps, a permutation π is generated on $[n]$ uniformly at each time t, and then assign the sub-interval $[\frac{\pi(i)-1}{m}, \frac{\pi(i)}{m})$ to the i^{th} path. The random seeds of the m-stratified sample paths are repellent to each other in $[0,1]^T$, hence, due to the choice of $f(i,e)$ in (5.3), the m-stratified sample paths are repellent to each other in the space of sample paths. This is summarized by Definition 20 and Algorithm 4. Compared to i.i.d. samples, the additional computational cost for generating stratified samples is negligible.

Definition 20. $\{X_i\}_{i \in [m]}$ *is called m-stratified samples if they are generated by Algorithm 4.*

5.3 Hypothesis Testing Using Stratified Samples

In this section, a statistical verification algorithm using stratified sampling is proposed to demonstrate the significant advantage of negatively correlated samples over independent samples in statistically verifying temporal logic specifications. As mentioned in Section 1, consider Probabilistic Computation Tree Logic formulas of the form $\mathbf{P}_{\sim p}\varphi$, where φ is a Linear Temporal Logic specification. When the sample paths $X_1, X_2, \ldots$ are drawn independently, the statistical verification prob-

Algorithm 4 m-stratified sampling

```
Require: Number of strata $m$, number of steps $T$, and initial state $s$
$t = 0$
for $i \in [m]$ do
    $X_i(0) = s$
end for
for $t = 1, \ldots, T-1$ do
    Take $\pi$ as a permutation of $[m]$
    for $i \in [m]$ do
        Take $E_i \sim \mathcal{U}_{[\frac{\pi(i)-1}{m}, \frac{\pi(i)}{m})}$
        $X_i(t+1) = f(X_i(t), E_i(t))$
    end for
end for
return $\{X_i\}_{i \in [m]}$
```

lem (2.3) can be solved efficiently with a sequential probability ratio test (SPRT), as mentioned in Section 5.3.

5.3.1 Properties of Stratified Samples

To implement the SPRT on m-stratified samples $\{X_i\}_{i\in[m]}$, consider the statistics

$$Y = \sum_{i=1}^{m} \varphi(X_i)/m. \tag{5.4}$$

The generation of the stratified samples in Algorithm 4 shows that

$$\mathbb{E}[Y] = \mathbb{E}\left[\sum_{i=1}^{m} \varphi(X_i)/m\right] = \mathbb{E}[\varphi(X_i)]. \tag{5.5}$$

In addition, for certain PCTL formulas φ, $\varphi(X_{1,i}), \ldots, \varphi(X_{m,i})$ can be generated negatively correlated, such that

$$\text{Var}[Y] \leq \text{Var}\left[\sum_{i=1}^{m} \varphi(X_i)/m\right] = \text{Var}[\varphi(X_i)]/m. \tag{5.6}$$

By the syntax of PCTL, φ is either of the form $\mathcal{X}\psi$ or $\psi_1 \mathcal{U}_{\leq T} \psi_2$, where ψ_1 and ψ_2 are directly checkable on the states of the Markov chain $\mathcal{M}$. Denote the set of states where ψ holds by

$$V_\psi = \{s \in [n] | \psi \in L(s)\} . \tag{5.7}$$

Assumption 3. *For a PCTL formula of the form $\varphi = \psi_1 \mathcal{U}_{\leq T} \psi_2$, it is assumed that*

(i) $V_{\psi_2} \subseteq V_{\psi_1}$*;*

(ii) The states of the Markov chain $\mathcal{M}$ are numbered such that $V_{\psi_1} = [n_1]$ and $V_{\psi_2} = [n_2]$ where $n_1 \geq n_2$.

Theorem 1. *With Assumption 3, let $\{X_i\}_{i \in [m]}$ be m-stratified samples from Markov chain $\mathcal{M}$ and φ be a probabilistic-operator-free PCTL formula with satisfaction probability p, then for any and $i \in [m]$,*

(i) $\mathbb{E}\left[\sum_{i=1}^m \varphi(X_i)/m\right] = \mathbb{P}\left[\varphi(X_i)\right]$*;*

(ii) Cov $\left[\varphi(X_i), \varphi(X_j)\right] \leq 0$ *for $i \neq j$.*

Now, the hypothesis testing problem (2.3) can be converted to

$$\begin{aligned} H_0' &: \mathbb{E}\left[Y\right] = p - \delta, \\ H_1' &: \mathbb{E}\left[Y\right] = p + \delta. \end{aligned} \tag{5.8}$$

In addition, the mean of m-stratified samples within each block are more concentrated than the mean of m independent samples with the same mean,

$$\begin{aligned} \mathrm{Var}\left[Y_i\right] &= \frac{1}{m^2}\mathrm{Var}\left[\sum_{j=1}^m \varphi(X_{j,i})\right] \\ &= \frac{1}{m}\mathrm{Var}\left[\varphi(X_{j,i})\right] + \frac{1}{m}\sum_{k=1,k\neq j}^m \mathrm{Cov}\left[\varphi(X_{j,i}), \varphi(X_{k,i})\right] \\ &\leq \frac{1}{m}\mathrm{Var}\left[\varphi(X_{j,i})\right] . \end{aligned} \tag{5.9}$$

Theorem 1 shows that compared to the mean m independent samples, the mean a group of m-stratified samples have the same mean, but smaller or at least equal variance. In addition, it shows that refining stratification always reduces the variance. Specifically, given an m-stratification, by refining each stratum into n strata, an mn-stratification can be derived. The new mn-stratified sampling algorithm will be no worse than the old m-stratified sampling algorithm.

Finally, there is no loss of statistical information by considering Y_i given by (5.4) instead of $(X_{1,i}, \ldots, X_{m,i})$.

Theorem 2. *Let $\pi(x_1, \ldots, x_m)$ be the joint probability mass function of $\varphi(X_1)$, $\ldots, \varphi(X_m)$, then the value of p only depends on $\sum_{i=1}^{m} \varphi(X_i)$.*

5.3.2 Sequential Probability Ratio Test

By Theorem 2, it suffices to consider Y_i to solve (2.3). Now given $Y^{(n)} = (Y_1, \ldots, Y_n) \subseteq \{0, 1/m, \ldots, 1\}$, an SPRT algorithm similar to (2.6) can be constructed,

$$\Lambda'(Y^{(n)}) = \Pi_{i=1}^{n} \frac{\pi_{H_1}(Y^{(n)})}{\pi_{H_0}(Y^{(n)})}. \tag{5.10}$$

where π_{H_1} and π_{H_0} are the probability mass function of Y_i under hypothesis H_0 and H_1 respectively.

However, unlike the i.i.d. case in (2.6), the exact form of π_{H_1} and π_{H_0} is hard to derive. Therefore, for simplicity, an asymptotic approach is taken via Central Limit Theorem. Let $\nu(Y^{(n)})$ be the empirical distribution given $Y^{(n)}$, then the Wald statistics converges to normal distribution $N(0,1)$ for large n

$$Z_n = \frac{\bar{Y}_i - \theta}{\sigma_i} \to N(0,1) \tag{5.11}$$

where $\theta = \mathbb{E}\left[Y\right]$ and

$$\bar{Y}_i = \frac{1}{i}\sum_{k=1}^{i} Y_k, \quad \sigma_i^2 = \frac{1}{i}\sum_{k=1}^{i}(Y_k - \bar{Y}_i)^2 \tag{5.12}$$

are the sample mean and sample variance respectively. Therefore, the probability ratio in (5.10) converges to

$$\Lambda'(Y^{(n)}) \to Ce^{-\frac{2(\bar{Y}_i - p)\delta}{\sigma_i^2}}, \quad n \to \infty, \tag{5.13}$$

for some normalizing constant C. In practice, this approximation is sufficiently accurate when the number of samples $n \geq 30$ and $\mathbb{E}\left[Y\right]$ is not close to the endpoints 0 and 1, since the converge of probability ratio (5.13) is faster. When $\mathbb{E}\left[Y\right]$ is close to 0 or 1, the distribution $\pi(y)$ of Y will become skew, and the convergence is slower [95, 96]. When the number of strata $m = 1$, the probability ratio (5.13) is equal (2.6) in large sample limit $n \to \infty$. Using (5.13), a sequential hypothesis testing algorithm can be constructed (Algorithm 5).

Algorithm 5 SPRT using stratified samples

```
Require: Number of strata $m$, Probability threshold $p$, Indifference Parameter $\delta$,
    Confidence level $\alpha, \beta > 0$, Minimal number of samples $N$
$r \leftarrow 0$
$\nu \leftarrow \{0, \ldots, 0\} \in \mathbb{Z}^{m+1}$
while true do
    $r \leftarrow r + 1$
    Take $m$-stratified samples $\{X_{1,r}, \ldots, X_{m,r}\}$
    $Y_r \leftarrow \sum_{i=1}^{m} \varphi(X_{i,r})$
    $\nu(Y_r) \leftarrow \nu(Y_r) + 1$
    if $r \geq N/m$ then
        $\mu_r \leftarrow \frac{\sum_{i=1}^{m+1} \frac{i-1}{m} \nu(i)}{\sum_{i=1}^{m+1} \nu(i)}$
        $\sigma_r^2 \leftarrow \left( \frac{\sum_{i=1}^{m+1} \left(\frac{i-1}{m}\right)^2 \nu(i)}{\sum_{i=1}^{m+1} \nu(i)} - \mu_r^2 \right) / r$
        if $\mu_r - p < -\frac{\sigma_r^2}{2\delta} \ln(\frac{1-\alpha}{\beta})$ then
            Return $H_0$
        else if $\mu_r - p > \frac{\sigma_r^2}{2\delta} \ln(\frac{1-\beta}{\alpha})$ then
            Return $H_1$
        end if
    end if
end while
```

5.4 Simulation

The sequential probability ratio test algorithm using stratified samples (Algorithm 5) is implemented on a small scale toy example and several more complicated benchmarks from [97]. In all the simulations, set the type I (2.4) error and type II error (2.5) to be 0.05, namely, the probability of the algorithm to make an error is always less than 5%. To guarantee sufficient accuracy of the probability ratio approximation (5.13), a minimal number of $N = 256$ samples is set for each run. The number of strata is taken to be $1, 2, 4, 8$. Accordingly, the minimal number of blocks are 256, 128, 64, 32; which are sufficient for large sample approximation (see Section 5.3.2) to hold.

Algorithm 5 is also compared with the sequential probability ratio test with independent samples proposed in [36, 37, 38], which is represented by SPRT in Table 5.2. The details of the simulation setups are given in Table 5.1.

Toy: A Discrete-Time Markov Chain of three states uniquely labeled by $\{1, 2, 3\}$ with probability transition matrix

$$\begin{bmatrix} 0.583 & 0.333 & 0.084 \\ 0.417 & 0.417 & 0.166 \\ 0.278 & 0.444 & 0.278 \end{bmatrix}.$$

Check

$$\mathbf{P}_{>p}(s \neq 2)\mathcal{U}_{[0,10]}(s = 1),$$

namely, whether the probability that a path avoids state 2 and finally returns back to state 1 within 10 steps is greater than p. The estimated probability for $(s \neq 2)\mathcal{U}_{[0,10]}(s = 1)$ to hold is 0.794956586 by the average of $1\,000\,000\,000$ i.i.d. samples. Therefore, the experiment is set for the following three cases

$$(p, \delta) = \begin{cases} (0.794956586 - 0.010002, 0.01), \\ (0.794956586 - 0.005002, 0.005), \\ (0.794956586 - 0.001002, 0.001). \end{cases}$$

where δ is the indifference parameter serving as an input to Algorithm 5.

One Die: A fair die modeled by a Discrete-Time Markov Chain of 13 states and 20 transitions proposed in [98]. Each state is labeled by only one of $s = 1, \ldots, s = 7$. Check

$$\mathbf{P}_{>p}\mathcal{F}_{[0,3]}(s > 6),$$

The estimated probability for $\mathbf{P}_{>p}\mathcal{F}_{[0,3]}(s > 7)$ to hold is 0.749987868 by the average of $1\,000\,000\,000$ i.i.d. samples. Therefore, the experiment is set for the following three cases

$$(p, \delta) = \begin{cases} (0.749987868 - 0.010002, 0.01), \\ (0.749987868 - 0.005002, 0.005), \\ (0.749987868 - 0.001002, 0.001). \end{cases}$$

Two Dice: The sum of two fair dice modeled by a Discrete-Time Markov Chain

Table 5.1: Summary of example models and testing formulas

Model	States	Transitions	Testing Formula
Toy	3	9	$\mathbf{P}_{>p}(s \neq 2)\mathcal{U}_{[0,10]}(s = 1)$
One Die	13	20	$\mathbf{P}_{>p}\mathcal{F}_{[0,3]}(s > 7)$
Two Dice	45	79	$\mathbf{P}_{>p}\mathcal{F}_{[0,4]}(s = 5)$
Election	1933	2557	$\mathbf{P}_{>p}\mathcal{F}_{[0,1]}(600 < s < 630)$

of 45 states and 79 transitions proposed in [98]. Similar to One Die, the states are either transient with at most two transitions with equal probability or sinks. Each state is labeled by only one of $s = 1, \ldots, s = 34$. Check $\mathbf{P}_{>p}\mathcal{F}_{[0,4]}(s = 5)$. The estimated probability for $\mathcal{F}_{[0,4]}(s = 5)$ to hold is 0.249983470 by the average of $1\,000\,000\,000$ i.i.d. samples. Therefore, the experiment is set for the following three cases

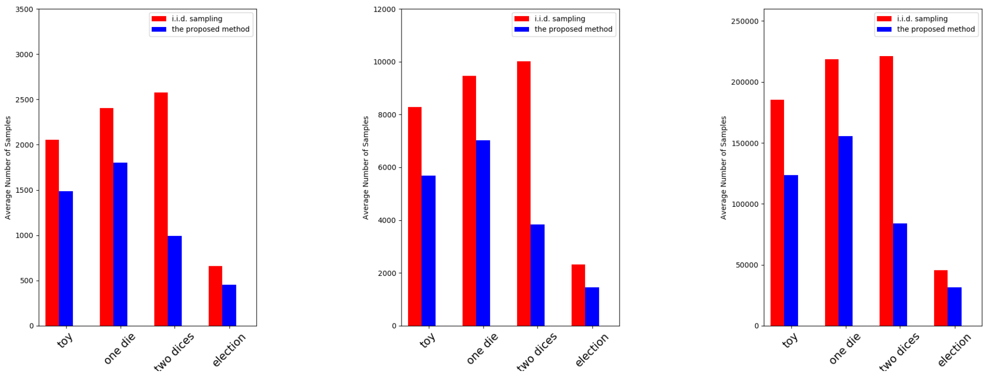
$$(p, \delta) = \begin{cases} (0.249983470 - 0.010002, 0.01), \\ (0.249983470 - 0.005002, 0.005), \\ (0.249983470 - 0.001002, 0.001). \end{cases}$$

Election: Synchronous leader election protocol of 4 processors and 5 candidates proposed in [99], which is modeled by a Discrete-Time Markov Chain of 1933 states and 2557 transitions. Check $\mathbf{P}_{>p}\mathcal{F}_{[0,1]}(600 < s < 630)$, where s is a numbering of the states. The estimated probability for $\mathcal{F}_{[0,1]}(600 < s < 630)$ to hold is 0.040002770 by the average of $1\,000\,000\,000$ i.i.d. samples. Therefore, the experiment is set for the following three cases

$$(p, \delta) = \begin{cases} (0.040002770 - 0.010002, 0.01), \\ (0.040002770 - 0.005002, 0.005), \\ (0.040002770 - 0.001002, 0.001). \end{cases}$$

The description of the simulation setups is summarized by Table 5.1. The simulation results for the above examples are shown in Table 5.2. The error probability and average sample size are derived by repeatedly running the algorithm for $10\,000$ to ensure statistical significance. The sample standard errors for the error probabilities and the average sample sizes are omitted in these tables for compactness.

The average sample size for Algorithm 5 for 1 stratum is approximately equal



(a) $\delta = 0.01$.

(b) $\delta = 0.005$.

(c) $\delta = 0.001$.

Figure 5.1: Summary of reduction in average sample sizes for the toy, one die, two dice and election examples for three choices of indifference parameter δ.

to the SPRT algorithm using independent samples. The former is always slightly larger than the latter, because there is a constraint on the minimal sample size. In all the cases, the actual type I error and type II error are controlled approximately below 0.05 with tolerable excess. These confirm that the large sample approximation used in Algorithm 5 is reasonable.

The reduction of sample size by stratification, as shown in Table 5.2a-5.2d, is visualized in Figure 5.1 below. The result shows that stratified sampling reduces the number of total samples (number of strata $\times$ number of blocks of stratified samples), compared to independent sampling. Specifically, Algorithm 5 for 8 strata reduces the number of total samples by $30\% - 60\%$ in the four examples.

Table 5.2: Average number of samples needed and error probabilities for SPRT with independent samples and Algorithm 5 for different strata sizes on the examples.

Case	1		2		3	
Strata	Samples	Error	Samples	Error	Samples	Error
SPRT	2054.4	4.68%	8276.8	4.38%	185310.9	2.74%
1	2275.5	5.13%	8708.4	4.66%	186778.3	2.80%
2	2283.6	5.33%	8684.3	4.41%	187172.8	2.76%
4	2083.6	5.63%	8038.4	4.33%	174372.5	2.99%
8	1485	4.93%	5692.3	4.55%	123723.0	2.79%

(a) Toy

Case	1		2		3	
Strata	Samples	Error	Samples	Error	Samples	Error
SPRT	2403.7	4.65%	9470.1	4.40%	218546.6	2.90%
1	2638.6	4.64%	9956.6	5.14%	221318.4	3.25%
2	1759.7	3.94%	6772.1	4.01%	149272.3	3.08%
4	1898.1	4.48%	7474.6	4.32%	162201.5	2.91%
8	1803.7	4.56%	7027.1	4.27%	155766.0	2.88%

(b) One Die

Case	1		2		3	
Strata	Samples	Error	Samples	Error	Samples	Error
SPRT	2573.5	4.82%	10019.2	4.54%	221101.3	2.67%
1	2605.9	3.67%	9878.5	3.86%	220478.8	3.17%
2	1753.0	4.14%	6702.0	4.54%	148054.3	2.81%
4	1180.1	4.02%	4499.1	4.38%	98349.0	3.03%
8	994.7	4.41%	3843.9	4.21%	84064.6	3.00%

(c) Two Dice

Case	1		2		3	
Strata	Samples	Error	Samples	Error	Samples	Error
SPRT	661.4	4.21%	2310.8	4.05%	45765.1	2.81%
1	586.4	1.33%	1976.1	1.91%	44506.4	2.20%
2	572.4	1.20%	1896.6	2.08%	42118.4	2.20%
4	535.8	1.57%	1758.4	2.22%	39154.0	2.43%
8	453.4	1.97%	1462.5	2.73%	31370.4	2.44%

(d) Election

Part II

Differential Privacy

CHAPTER 6

DIFFERENTIAL PRIVACY IN DISTRIBUTED SYSTEMS

This chapter focuses on a general framework of distributed control systems in which agents share information based on randomized mechanisms, and sets a base for the next two chapters. The formulation of the system is given in Section 6.1, the definition for ε-differential privacy in the system is given in Section 6.2, metric of tracking performance in Section 6.3, and the unbiased estimators of the private data and its accuracy in Section 6.4.

6.1 Linear Distributed Systems with Randomized Communication

Let us consider a linear distributed control system with N agents in which each agent will use a randomized mechanism to share information on a finite time horizon $T > 0$. A schematic diagram of the system is shown in Figure 6.1.

The dynamics of an individual agent is influenced by the *actual* states of other agents. For example in a distributed traffic control scenario [100], any particular agent's speed and choice of the route are influenced by the state of other agents as they share a common resource, the roadways. By explicitly exchanging information about their states, the agents could achieve better performance (routing delays), but at the same time, by sharing exact information about their states they may give away too much information about their private data. Thus the agents choose to share only noisy versions of their state using a randomized mechanism. Formally, the state evolution $x_i \in \mathcal{X} = \mathbb{R}^n$ of agent i is modeled as a discrete time dynamical system:

$$x_i(t+1) = Ax_i(t) + v_i(t) + \frac{c}{N} \sum_{j \in [N]} x_j(t), \tag{6.1}$$

where (a) $x_i(t)$ is the state of agent i; (b) v_i is the local control input; (c)$c \in$

$\mathbb{R}$ is a coupling constant capturing the aggregate influence of the other agents, for example, congestion. The aggregated state of the system is denoted by $x = (x_1, x_2, \ldots, x_N) \in \mathcal{X}^N$. Besides, each agent is given *a priori* a preference (or target) $p_i(t)$, namely a sequence of waypoints.

As alluded to above, to keep the communications private, agent i adds noise n_i to its state and reports this noisy state $\tilde{x}_i$ to the server:

$$\tilde{x}_i(t) = x_i(t) + n_i(t). \tag{6.2}$$

Let the noisy state live in the same space as the actual state, $\tilde{x}_i \in \mathcal{X} = \mathbb{R}^n$.

The goal of each agent i is to track the preference. To achieve this, the agents use a feedback control $v_i(t)$ based on the information $u_i(t) = -\frac{c}{N}\sum_{j\in[N]} \tilde{x}_j$ received from the server, where $\tilde{x}_i$ is the reported state of agent i. Due to privacy consideration, the reported states are not precisely the states of the agents, but random variables that approximate the real states. Here, consider the following linear feedback control law for the agents,

$$v_i(t) = K'(x_i(t) - p_i(t+1)) + (I - A)p_i(t+1) - u_i(t), \tag{6.3}$$

where the $K'(x_i(t) - p_i(t+1))$ is a linear feedback term of the tracking error, $(I - A)p_i(t+1)$ is an additive term to move the equilibrium of $x_i(t)$ to $p_i(t+1)$, and $-u_i(t)$ tries to cancel the effect of the aggregate state. Thus,

$$\begin{align}
\tilde{x}_i(t) &= x_i(t) + n_i(t), \tag{6.4}\\
u_i(t) &= \frac{c}{N}\sum_{j\in[N]} \tilde{x}_j, \tag{6.5}\\
x_i(t+1) &= Kx_i(t) + (I - K)p_i(t+1) \nonumber\\
&\qquad - u_i(t) + \frac{c}{N}\sum_{j\in[N]} x_j(t), \tag{6.6}
\end{align}$$

where $K = K' + A \in \mathbb{R}^{n\times n}$ is the closed-loop dynamics matrix and $c \in \mathbb{R}$ is a coupling constant. If another linear feedback control is used, this analysis can be applied with straightforward modifications.

For an individual agent, the combination of the initial state and the sequence of preferences $d_i = (x_i(0), p_i(1), \ldots, p_i(T-1))$ is referred to as the *private data* of the agent. Similarly, the *private data set* of the system is the ordered collection $D = \{d_i | i \in [N]\}$ of N elements, such that the ith element in D is d_i. The set of all private data is denoted by $\mathcal{D}$. In addition, refer to (a) the sequence of aggregated states $x(0), x(1), \ldots, x(T-1)$ generated by the system as the *trajectory* of the system, and (b) the sequence of reported states $\tilde{x}(0), \tilde{x}(1), \ldots, \tilde{x}(T-1)$ generated by the system as the *observation sequence* of the system. For a private data set D of the system, denote $O_D = \{\tilde{x}(t)\}_{t=0}^{T-1}$ as the observation sequence up to time T that takes values in $\mathcal{X}^{NT}$.

Combining the above equations, the closed-loop dynamics of agent i is:

$$x_i(t+1) = Kx_i(t) + (I-K)p_i(t+1) - \frac{c}{N}\sum_{j\in[N]} n_j(t). \tag{6.7}$$

Agent i's state at time t can be written as a function of its preference sequence $\{p_i(s)\}_{s\in[t]}$ and the sequence $\{n_i(s)|i \in [N], s \in [t]\}$ of noise vectors added in all previous rounds. Iteratively applying (6.7) gives

$$\begin{aligned} x_i(t) =& K^t x_i(0) + \sum_{s=1}^{t} K^{t-s}(I-K)p_i(s) \\ & - \frac{c}{N}\sum_{s=0}^{t-1} K^{t-s-1} \sum_{j\in[N]} n_j(s). \end{aligned} \tag{6.8}$$

By (6.8), it is obvious that the following statement holds.

Remark 2. *Given the private data set D, the system trajectory $\{x(t)\}_{t<T}$ is uniquely determined by the value of the sequence of reported states $O_D = \{\tilde{x}(t)\}_{t<T}$.*

Denote $\rho(D, O) = \{x(t)\}_{t<T}$ as the trajectory corresponds to data set D and observation O and $\rho(D, O)(t) = x(t)$ as the state of the trajectory at time t. Formally, the randomized mechanism $\mathcal{M}$ has two components: (i) a discrete-time stochastic process, $\{n(t)\}_{t<T}$ and (ii) a *deterministic function* that maps the private data set D and the outputs of the stochastic process to an observation sequence $\{\tilde{x}(t)\}_{t<T}$. In the rest of the part, it is assumed that the stochastic process $\{n(t)\}_{t<T}$ is well behaved such that every $n(t)$ has absolute continuous probability distribution function.

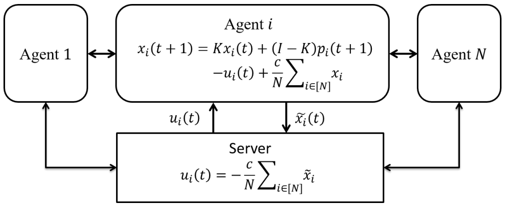

Agent 1
Agent i
$x_i(t+1) = Kx_i(t) + (I-K)p_i(t+1)$
$-u_i(t) + \frac{c}{N}\sum_{i\in[N]} x_i$
Agent N
$u_i(t)$
$\tilde{x}_i(t)$
Server
$u_i(t) = -\frac{c}{N}\sum_{i\in[N]} \tilde{x}_i$

Figure 6.1: Diagram of a distributed control system.

6.2 Differential Privacy of Distributed Control Systems

The standard definition of differential privacy in the context of databases [49, 50] does not consider the metric on the data. Here, in the context of dynamical systems, the metric version of *differential privacy* used in [61] is adopted.

Definition 21. *For two data sets $D = \{d_i\}_{i\in[N]}$ and $D' = \{d_i'\}_{i\in[N]}$, the* distance *between the sets is $d(D, D') = \sum_{i\in[N]} \|d_i - d_i'\|_1$.*

Definition 22. *Given a time horizon $T > 0$ and a parameter $\epsilon > 0$, a randomized mechanism $\mathcal{M} : D \to O_D$ is ϵ-Differentially Private up to time $T-1$, if for any subset $\mathcal{O} \subseteq \mathcal{X}^{NT}$ and any two data sets D, D', the inequality*

$$\mathbb{P}\left[O_D \in \mathcal{O}\right] \leq e^{\epsilon d(D,D')}\mathbb{P}\left[O_{D'} \in \mathcal{O}\right] \tag{6.9}$$

holds, where the random variables O_D and $O_{D'}$ are the observation sequences generated by the two data sets D and D'.

Remark 3. *If the system is ϵ-Differentially Private up to time $T-1$, then it is ϵ-Differentially Private up to any time $S < T$. Taking $\mathcal{O} = \mathcal{O}' \times \mathcal{X}^{T-S}$, where $\mathcal{O}' \subseteq \mathcal{X}^S$, gives the condition for ϵ-Differential Privacy up to time S.*

Roughly speaking, the definition above requires that the probabilities of getting the same observation sequence are close depending on the distance between the two data sets. In other words, the probability that a small change in the private data is detected from the observation sequence is very low.

The privacy of the system increases as ϵ decreases. For $\epsilon \to \infty$, all randomizing mechanisms are ϵ-Differentially Private; for $\epsilon = 0$, only the mechanisms that generate identical observation sequences will be ϵ-Differentially Private.

6.3 Influence on the Dynamics of the System

Adding noise to the system affects the dynamics of the system. In this section, the trade-off between Differential Privacy and the mean-square tracking error is studied. For a ϵ-differentially private mechanism $\mathcal{M}$ and a private data set D as discussed in Section 6.1, the mean-square tracking error is used to define a cost function for agent i up to time $T-1$ by

$$\mathrm{cost}_{\epsilon,D,i} = \mathbb{E}\left[\sum_{t=1}^{T-1} \|x_i(t) - p_i(t)\|_2^2\right]. \tag{6.10}$$

Obviously, the cost functions increase with time T. Let $\{\overline{x}(0), \ldots, \overline{x}(T-1)\}$ be the aggregate trajectory of the system with data set D and no noise, ***i.e.***, $n_i(t) = 0$ for all t. Define $\overline{\mathrm{cost}}_{D,i} = \sum_{t=1}^{T-1} \|\overline{x}_i(t) - p_i(t)\|_2^2$ be the cost associated with agent i. The *cost of privacy* of mechanism $\mathcal{M}$ is defined as the supremum in the change of single agent's cost over all data sets:

$$\Delta(\epsilon, T) = \sup_{\substack{i\in[N]\\ D\in\mathcal{D}}} (\mathrm{cost}_{\epsilon,D,i} - \overline{\mathrm{cost}}_{D,i}), \tag{6.11}$$

which is called $\Delta(\epsilon, T)$ the *cost of privacy* (or CoP). It increases when larger noise is added to the system.

6.4 Accuracy of unbiased estimators

Consider the unbiased estimator

$$\hat{D} = \{(\hat{x}_i(0), \hat{p}_i(1), \ldots, \hat{p}_i(T-1)) | i \in [N]\} \tag{6.12}$$

of the private data from a sequence of reported states O_D. Since $\hat{D}$ is a function of O_D, thus for any θ,

$$f(\theta|D) \leq e^{\epsilon d(D-D')} f(\theta|D') \tag{6.13}$$

where $f(\theta|D)$ and $f(\theta|D')$ are the probability distribution functions of $\hat{D}$ given the private data D and D', respectively.

There are multiple ways to measure the accuracy of the estimator $\hat{D}$, including variance, high-order moment and entropy. In this work, the (Shannon) entropy $H(\hat{D})$ is used to measure the amount of information that can be derived from the estimation. It decreases when the p.d.f. of the estimator becomes sharper, and *vice versa*. It will be shown in Section 8 that if the system is ϵ-Differentially Private, then there is a lower bound on $H(\hat{D})$ and the minimum is achieved by adding Laplace noise.

CHAPTER 7

DIFFERENTIALLY PRIVATE AND TRACKING

In this section, the impact of differential privacy is studied on the tracking performance of a linear distributed control system. It can be shown that there is a trade-off between the level of differential private of the agents' private data and the tracking performance of the system.

7.1 Sensitivity and Differential Privacy

For a fixed data set D, each observation sequence $\{\tilde{x}(t)\}_{t<T}$ corresponds to a unique trajectory $\{x(t)\}_{t<T}$ independent of the precise design of mechanism $\mathcal{M}$. A differentially private mechanism can be proposed for linear distributed control systems using the idea of sensitivity.

Definition 23. *The* sensitivity *of randomized mechanism at time* t *is defined as the supremum* ℓ^1*-norm between trajectories corresponding to the same observation sequence and data sets:*

$$S(t) = \sup_{\substack{\mathrm{adj}(D,D') \\ O \in \mathcal{X}^{NT}}} \frac{\|\rho(D,O)(t) - \rho(D',O)(t)\|_1}{d(D,D')}. \tag{7.1}$$

If the sensitivity is finite, the following lemma established in [75] suggest that by adding a sequence of Laplace noise from a distribution with parameter $S(t)/\epsilon$ at each round t, the mechanism is differentially private.

Lemma 4. *For any time bound* T *and privacy parameter* $\epsilon > 0$*, for* $M_t \triangleq TS(t)/\epsilon$*, a randomized mechanism is* ϵ*-differentially private up to time* T*, if the noise* $\{n(t)\}_{t<T}$ *is independent and follows* $\{\mathrm{Lap}(M_0, nN), \mathrm{Lap}(M_1, nN), \ldots, \mathrm{Lap}(M_{T-1}, nN)\}$.

Proof. Fix any pair of data sets D, D', and any set of observation sequences $\mathcal{O} \subseteq \mathcal{X}^{NT}$. Denote f_D and $f_{D'}$ as the probability density functions of random process

$\{x(t)\}_{t=0}^{T}$ with data sets D and D' respectively. Define the sets of trajectories $A \triangleq \{\rho(D, O) : O \in \mathcal{O}\}$ and $A' \triangleq \{\rho(D', O) : O \in \mathcal{O}\}$ respectively. Thus,

$$\frac{\mathbb{P}\left[O_D \in \mathcal{O}\right]}{\mathbb{P}\left[O_{D'} \in \mathcal{O}\right]} = \frac{\int_{\alpha \in A} f_D(\alpha) d\mu}{\int_{\alpha' \in A'} f_{D'}(\alpha') d\mu'}. \tag{7.2}$$

First, define a correspondence B between the sets A and A', such that for $\alpha \in A$ and $\alpha' \in A'$, $B(\alpha) = \alpha'$ if they have the same observation sequence up to time T. From Proposition 2, thus $\alpha = \rho(D, O)$ and $\alpha' = \rho(D', O)$ are both unique. Therefore B is a bijection. The probability of the sets A and A' are related via the bijection B,

$$\begin{aligned} \int_{\alpha' \in A'} f_{D'}(\alpha') d\mu' &= \int_{B(\alpha) \in A'} f_{D'}(B(\alpha)) d\mu \\ &= \int_{\alpha \in A} f_{D'}(B(\alpha)) d\mu. \end{aligned} \tag{7.3}$$

For a data set D, the trajectory $\alpha = \{x(t)\}_{t<T}$ is uniquely defined by the noise sequence $\{n(t)\}_{t<T}$, which follows $\{\mathrm{Lap}(M_0, nN), \mathrm{Lap}(M_1, nN), \ldots, \mathrm{Lap}(M_{T-1}, nN)\}$. For any observation $O \in \mathcal{O}$ and trajectory $\alpha = \rho(D, O)$, denote $O_i^{(k)}(t)$ as the kth entry of the observation vector $\tilde{x}_i(t)$, and $\alpha_i^{(k)}(t)$ as the kth entry of the state vector $x_i(t)$. Then the probability density of trajectory α is

$$f_D(\alpha) = \prod_{\substack{i \in [N],\ k \in [n] \\ t < T}} f_L(O_i^{(k)}(t) - \alpha_i^{(k)}(t), M_t), \tag{7.4}$$

where $f_L(\cdot, \lambda)$ is the probability density of scalar Laplace distribution $\mathrm{Lap}(\lambda)$. Similarly, for data set D', the probability density function is the same

$$f_{D'}(\alpha) = \prod_{\substack{i \in [N],\ k \in [n] \\ t < T}} f_L(O_i^{(k)}(t) - \alpha_i^{(k)}(t), M_t). \tag{7.5}$$

Then, the distance between the trajectories $\alpha = \rho(D, O)$ and $B(\alpha)$ is bounded with the sensitivity $S(t)$. By the Definition 23, thus

$$\|\rho(D, O)(t) - \rho(D', O)(t)\|_1 \leq S(t) d(D, D'). \tag{7.6}$$

By definition of ℓ^1-norm:

$$\begin{aligned}\sum_{i=1}^{N}\sum_{k=1}^{n}|\alpha_i^{(k)}(t) - B(\alpha)_i^{(k)}(t)| &= \|\rho(D,O)(t) - \rho(D',O)(t)\|_1 \\ &\leq S(t)d(D,D').\end{aligned} \tag{7.7}$$

For scalar Laplace distribution $\mathrm{Lap}(\lambda)$ and any $x, x' \in \mathbb{R}$, thus $\frac{f_L(x,\lambda)}{x',\lambda} \leq e^{\frac{|x-x'|}{\lambda}}$. This property gives

$$\begin{aligned}&\prod_{i\in[N]k\in[n]} \frac{f_L(O_i^{(k)}(t) - \alpha_i^{(k)}(t), M_t)}{f_L(O_i^{(k)}(t) - B(\alpha)_i^{(k)}(t), M_t)} \\ \leq &\prod_{i\in[N],k\in[n]} e^{\frac{|(O_i^{(k)}(t) - \alpha_i^{(k)}(t), M_t) - (O_i^{(k)}(t) - B(\alpha)_i^{(k)}(t)|}{M_t}} \\ = &\exp\left(\sum_{i\in[N],k\in[n]} \frac{|\alpha_i^{(k)}(t) - B(\alpha)_i^{(k)}(t)}{M_t}\right) \\ \leq &\exp\left(\frac{S(t)d(D,D')}{M_t}\right).\end{aligned} \tag{7.8}$$

Combining (7.2)-(7.8) gives

$$\begin{aligned}\frac{\mathbb{P}\left[O_D \in \mathcal{O}\right]}{\mathbb{P}\left[O_{D'} \in \mathcal{O}\right]} \leq &\prod_{t=0}^{T-1} \exp\left(\frac{S(t)}{M_t}\right) \\ \leq &\exp\left(\sum_{t=0}^{T-1} \frac{S(t)d(D,D')}{M_t}\right).\end{aligned} \tag{7.9}$$

If the sequence of M_t satisfies $\sum_{t\in[T]} \frac{S(t)}{M_t} \leq \epsilon$, then $\frac{\mathbb{P}[O_D \in \mathcal{O}]}{\mathbb{P}[O_{D'} \in \mathcal{O}]} \leq \exp\left(\epsilon d(D,D')\right)$. Thus the mechanism is ϵ-differentially private. □

If a bound on the sensitivity $S(t)$ of the linear distributed system (6.4)-(6.6) is established at each round t, a differentially private mechanism can be implemented. To represent the dynamics of the aggregated system with N agents, define two $nN \times nN$ matrices

$$\begin{aligned}\mathbf{K} &= I_N \otimes K, \\ \mathbf{C} &= \mathbf{1}_N \otimes \frac{cI_n}{n},\end{aligned} \tag{7.10}$$

where I_N is the $N \times N$ identity matrix, $\mathbf{1}_N$ is the $N \times N$ matrix with all elements being 1, and $\otimes$ denotes the Kronecker product. Combining (6.4)-(6.6) for all the N agents without unrolling the $u_i(t)$ terms gives

$$\begin{aligned} x(t+1) &= \mathbf{K}x(t) + (I - \mathbf{K})p(t+1) + \mathbf{C}x(t) - u(t) \\ &= (\mathbf{K} + \mathbf{C})x(t) + (I - \mathbf{K})p(t+1) - u(t). \end{aligned} \tag{7.11}$$

Iteratively applying the above equation gives

$$\begin{aligned} x(t) = (\mathbf{K} + \mathbf{C})^t x(0) - \sum_{s=0}^{t-1} (\mathbf{K} + \mathbf{C})^{t-s} u(t) \\ + \sum_{s=1}^{t} (\mathbf{K} + \mathbf{C})^{t-s} (I - \mathbf{K}) p(s). \end{aligned} \tag{7.12}$$

To prove the following theorem, fix two private data sets D and D' and compute the difference between the two trajectories. Recall that D and D' are identical except the preference of one agent (i). Then, the difference between the two trajectories has two components: (1) the change in agent i's state, and (2) the sum of changes in other agents' state. The sensitivity is then computed as a bound of the sum of the above two components.

Theorem 9. *For the linear distributed control system, for all $t \in \mathbb{N}$ the sensitivity $S(t) \leq \kappa(t)$, where κ is defined as*

$$\kappa(t) \triangleq \|G^t - K^t\|_1 + \|K^t\|_1 + \|H\|_1 \sum_{s=0}^{t-1} (\|G^s - K^s\|_1 + \|K^s\|_1), \tag{7.13}$$

with $G \triangleq cI + K$ and $H \triangleq I - K$.

Proof. Take a pair of private data sets D and D', and a sequence of observations $O = \{\tilde{x}(t)\}_{t<T}$. By (6.5), the input $\{u(t)\}_{t<T}$ is also fixed. Then, by (7.12)

$$\begin{aligned} \|\rho(D, O)(t) - \rho(D', O)(t))\|_1 = \|(\mathbf{K} + \mathbf{C})^t (x(0) - x'(0)) \\ + \sum_{s=1}^{t} (\mathbf{K} + \mathbf{C})^{t-s} (I - \mathbf{K})(p(s) - p'(s))\|_1. \end{aligned} \tag{7.14}$$

Expanding the term $(\mathbf{K}+\mathbf{C})^s$ on the right-hand side of (7.14) gives

$$(\mathbf{K}+\mathbf{C})^s = \left(\begin{bmatrix} K & & \\ & \ddots & \\ & & K \end{bmatrix} + \frac{c}{N} \begin{bmatrix} I & \dots & I \\ \vdots & \ddots & \vdots \\ I & \dots & I \end{bmatrix} \right)^s . \tag{7.15}$$

The matrix $(\mathbf{K}+\mathbf{C})$ has two types of blocks: (1) $K+\frac{c}{N}I$ as the diagonal blocks and (2) $\frac{c}{N}I$ as the off-diagonal blocks. As K and I are commutative, applying binomial expansion of the (7.15) and after some lengthy but elementary linear algebra the product matrix $(\mathbf{K}+\mathbf{C})^s$ becomes

$$(\mathbf{K}+\mathbf{C})^s = \begin{bmatrix} P_s & Q_s & \dots & Q_s \\ Q_s & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & Q_s \\ Q_s & \dots & Q_s & P_s \end{bmatrix} \text{where} \tag{7.16}$$

$$Q_s = \frac{1}{N}(G^s - K^s), \text{ and } P_s = Q_s + K^s, \tag{7.17}$$

where $G \triangleq cI + K$. (7.16) implies that

$$(\mathbf{K}+\mathbf{C})^s(I-\mathbf{K}) = \begin{bmatrix} P'_s & Q'_s & \dots & Q'_s \\ Q'_s & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & Q'_s \\ Q'_s & \dots & Q'_s & P'_s \end{bmatrix} \tag{7.18}$$

where $Q'_s = Q_sH$, $P'_s = Q'_s + K^sH$ and $H = I - K$. With (7.16) and (7.18), the right-hand side of (7.14) is bounded. By definition of adjacency, $\mathrm{adj}(D, D')$ if and only if there exists some $i \in [N]$, such that for any $s \leq t$ and all $j \neq i$, $p_j(s) - p'_j(s) = 0$. That is, for any $s \leq t$,

$$p(s) - p'(s) = \Big[0, \quad \dots, 0, [p_i(s) - p'_i(s)]^\top, 0, \dots, 0\Big]^\top , \tag{7.19}$$

has n non-zero entries corresponding to the preferences of some agent i, and all other entires are 0. Then, $(\mathbf{K}+\mathbf{C})^s(p(s)-p'(s))$ is a vector with the ith block as $P_s(p_i(s)-p'_i(s))$ and other blocks as $Q_s(p_i(s)-p'_i(s))$. Similarly $(\mathbf{K}+\mathbf{C})^s(I-K)(p(s)-p'(s))$ is a vector with the ith block as $P'_s(p_i(s)-p'_i(s))$ and other blocks as $Q'_s(p_i(s)-p'_i(s))$. Therefore, the term inside the norm on the right-hand side of (7.14) is a vector where the ith block is

$$P_t(x_i(0) - x'{}_i(0)) + \sum_{s=1}^{t} P'_{t-s}(p_i(s) - p'{}_i(s)), \tag{7.20}$$

and all the other $N-1$ components are

$$Q_t(x_i(0) - x'_i(0)) + \sum_{s=1}^{t} Q'_{t-s}(p_i(s) - p'_i(s)). \tag{7.21}$$

Substituting (7.20) and (7.21) into (7.14) and combining with $\|x_i(0) - x'_i(0)\|_1 + \|(p_i(s) - p'_i(s)\|_1 = d(D, D')$ give

$$S(t) \leq (N-1)(\|Q_t\|_1 + \sum_{s=0}^{t-1} \|Q'_s\|_1) + \|P_t\|_1 + \sum_{s=1}^{t} \|P'_s\|_1. \tag{7.22}$$

Using (7.17), P_s, P'_s is represented by Q_s, Q'_s, K and H. Therefore,

$$\begin{aligned} S(t) \leq\ & (N-1)(\|Q_t\|_1 + \sum_{s=0}^{t-1} \|Q'_s\|_1) + \|Q_t\|_1 + \|K^t\|_1 \\ & + \sum_{s=1}^{t} \|Q'_s\| + \sum_{s=1}^{t} \|K^s\|_1 \|H\|_1 \\ =\ & N(\|Q_t\|_1 + \sum_{s=0}^{t-1} \|Q'_s\|_1) + \|K^t\|_1 + \|H\|_1 \sum_{s=1}^{t} \|K^s\|_1 \end{aligned} \tag{7.23}$$

Again from (7.17), substitute Q_s and Q'_s by H, G and K,

$$\begin{aligned} S(t) \leq\ & \|G^t - K^t\|_1 + \|K^t\|_1 \\ & + \|H\|_1 \sum_{s=1}^{t} (\|G^s - K^s\|_1 + \|K^s\|_1) \\ =\ & \kappa(t). \end{aligned} \tag{7.24}$$

□

Remark 4. *The upper bound on the sensitivity at time t, $\kappa(t)$ has two components:*

(a) $\|K^t\|_1 + \|H\|_1 \sum_{s=1}^{t} \|K^s\|_1$ overapproximates the change in agent i's state (x_i) if its own preference changes at each time up to t, and

(b) $\|G^t - K^t\|_1 + \|H\|_1 \sum_{s=0}^{t-1} \|G^s - K^s\|_1$ overapproximates the sum of the

changes in other agents' state given agent i's preference changes upto t.

Remark 5. *$\kappa(t)$ is independent of the number of agents (N). It only depends on matrix K, the coupling constant c and time t. K is specified by the individual's control function, which assumes to be stable. The more stable the matrix K is, the faster $\|K^t\|_1$ decays to 0. The coupling constant c quantifies the influence of the aggregate on each agent. The matrix $G = cI + K$ captures the combined dynamics under the influence of the environment and the dynamics of the individual agents. The weaker the physical coupling, the smaller $\|G^t\|_1$. Therefore, as the dynamics of the individual agents become more stable or the physical coupling between agents becomes weaker, the sensitivity of the system decreases.*

Remark 6. *The dependence of $\kappa(t)$ on time t changes based on the stability of the K and G matrices. If G and K are stable, $\kappa(t)$ converges to a constant as $t \to \infty$. Otherwise, $\kappa(t)$ grows exponentially with t.*

Theorems 9 immediately suggest a noise-adding mechanism which guarantees differential privacy of the distributed linear control system.

Example 4. *Apply the strategy explained above to a system with $K = \frac{1}{5}I_2$. $G = (c + \frac{1}{5})I_2$. By Theorem 9, the sensitivity bound is*

$$S(t) \leq \kappa(t) = \frac{4 + 20c}{20 - 25c} + \frac{16 - 45c}{20 - 25c}\left(c + \frac{1}{5}\right)^t \tag{7.25}$$

As stated in Remark 5, the sensitivity bound is independent of N. If G is stable, that is $|c + \frac{1}{5}| \leq 1$, the sensitivity $S(t)$ is bounded and converges to a constant as $t \to \infty$. Otherwise, if $|c + \frac{1}{5}| > 1$, $\kappa(t)$ diverges. The parameter of the Laplace distribution is chosen to be $M_t = \frac{\kappa(t)T}{\epsilon}$. By Lemma 4, the system guarantees ϵ-differential privacy upto time T for arbitrary T.

7.2 Cost of Privacy in Linear Distributed Control

In this section, the cost of privacy for the randomizing mechanism is studied compared to a perfectly observable (noise free) system using the same controller. First, from (6.7), the tracking behavior of the system depends on the matrix K.

Remark 7. *Taking expectation on both sides of* (6.7) *gives $\mathbb{E}\left[x_i(t) - p_i(t)\right] = K\mathbb{E}\left[x_i(t-1) - p_i(t)\right]$. If the closed-loop matrix K is Hurwitz, the state of each agent converges to the preference in expectation.*

The cost of privacy varies depending on the proper of K.

Theorem 10. *The cost of privacy of the ϵ-differentially private mechanism $\mathcal{M}$ of Lemma 4 and Example 4 is of order $O(\frac{T^3}{N\epsilon^2})$ if the matrix K is Hurwitz. Otherwise, it grows exponentially with T.*

Proof. Given the ϵ-differentially private mechanism $\mathcal{M}$, the perfectly observable system is obtained by setting the noise values to be 0. Denote by $\bar{x}_i(t)$ the state of agent i for the perfectly observable system at time t. From (6.8), by fixing $n_i(t) = 0$,

$$\bar{x}_i(t) = K^t p_i(0) + \sum_{s=1}^{t} K^{t-s}(I - K)p_i(s).$$

Define a $n \times nN$ matrix $\mathbf{B} \triangleq \frac{c}{N}[I, \ldots, I]$. Again from (6.8), the state of an individual agent i is

$$x_i(t) = \bar{x}_i(t) - \sum_{s=0}^{t-1} K^{t-s-1}\mathbf{B}n(s).$$

The cost of the mechanism $\mathcal{M}$ can be written as

$$\begin{aligned}
\text{cost}_{\epsilon,D,i} &= \mathbb{E}\left[\sum_{t=1}^{T-1} \|x_i(t) - p_i(t)\|_2^2\right] \\
= \ & \mathbb{E}\left[\sum_{t=1}^{T-1} \|\bar{x}_i(t) - \sum_{s=0}^{t-1} K^{t-s-1}\mathbf{B}n(s) - p_i(t)\|_2^2\right] \\
= \ & \sum_{t=1}^{T-1} \mathbb{E}\left[\|\bar{x}_i(t) - p_i(t)\|_2^2 + \|\sum_{s=0}^{t-1} K^{t-s-1}\mathbf{B}n(s)\|_2^2\right. \\
& \left. -2(\bar{x}_i(t) - p_i(t))^\top \sum_{s=0}^{t-1} K^{t-s-1}\mathbf{B}n(s)\right]
\end{aligned}$$

The first term on the right-hand side is the cost of the system with perfect observations, that is, $\overline{\text{cost}}_{D,i}$. The last term on the right-hand side is the expectation of a linear combination of zero-mean noise terms, and therefore, equals 0. By Definition,

$$\begin{aligned}
\Delta(\epsilon, T) &= \sup_{D,i}[\text{cost}_{\epsilon,D,i} - \overline{\text{cost}}_{D,i}] \\
&= \sum_{t=1}^{T-1} \mathbb{E}\left[\|\sum_{s=0}^{t-1} K^{t-s-1}\mathbf{B}n(s)\|_2^2\right]
\end{aligned} \tag{7.26}$$

In the mechanism $\mathcal{M}$, for different time steps s, τ, the noise $n(s)$ and $n(\tau)$ are independent. The right-hand side of (7.26) reduces to

$$\sum_{t=1}^{T-1} \mathbb{E}\left[\sum_{s=0}^{t-1} n(s)^\top \mathbf{B}^\top (K^{t-s-1})^\top K^{t-s-1} \mathbf{B} n(s)\right].$$

Denote $n^{(k)}(s)$, $k \in [nN]$, be the kth element of the vector $n(s)$. It follows that (a) for $k \neq j \in [nN]$, $\mathbb{E}\left[n^{(k)}(s) n^{(j)}(s)\right] = 0$, and (b) for any $k \in [nN]$, $\mathbb{E}\left[n^{(k)}(s) n^{(k)}(s)\right] = 2M_s^2$. Thus, the above expression is reduced to

$$\sum_{t=1}^{T-1} \sum_{s=0}^{t-1} 2M_s^2 \mathrm{Tr}(\mathbf{B}^\top (K^{t-s-1})^\top K^{t-s-1} \mathbf{B}), \tag{7.27}$$

where $\mathrm{Tr}(A)$ stands for the trace of matrix A. Recall that $\mathbf{B} = \frac{c}{N}[I, \ldots, I]$. It follows that

$$\begin{aligned} &\mathrm{Tr}(\mathbf{B}^\top (K^{t-s-1})^\top K^{t-s-1} \mathbf{B}) \\ =&\frac{c^2}{N} \mathrm{Tr}((K^{t-s-1})^\top K^{t-s-1}) = \frac{c^2}{N} \|K^{t-s-1}\|_2^2. \end{aligned}$$

Substituting the above equation into (7.27) yields

$$\Delta(\epsilon, T) = \frac{2c^2}{N} \sum_{t=1}^{T-1} \sum_{s=0}^{t-1} M_s^2 \|K^{t-s-1}\|_2^2$$

Interchanging the order of summation gives

$$\begin{aligned} \Delta(\epsilon, T) =& \frac{2c^2}{N} \sum_{s=0}^{T-2} \sum_{t=s+1}^{T-1} M_s^2 \|K^{t-s-1}\|_2^2 \\ =& \frac{2c^2}{N} \sum_{s=0}^{T-2} M_s^2 \sum_{t=0}^{T-s-2} \|K^t\|_2^2. \end{aligned} \tag{7.28}$$

Recall that in Lemma 4, $M_s = \frac{T\kappa(s)}{\epsilon}$. Combining this with (7.28),

$$\Delta(\epsilon, T) = \frac{2c^2(T-1)^2}{N\epsilon^2} \sum_{s=0}^{T-2} \kappa(s)^2 \sum_{t=0}^{T-s-2} \|K^t\|_2^2.$$

From the above expression it is clear $\Delta(\epsilon, T)$ is inversely proportional to N and ϵ^2. As the matrix K is Hurwitz, $\sum_{t=0}^{T-s-2} \|K^t\|^2$ converges to some constant as

$T \to \infty$. By Remark 6, if G is stable then $\kappa(s)$ converges to some constant as $s \to \infty$, $\sum_{s=0}^{T-2} \kappa(s)^2$ grows linearly with T, thus $\Delta(\epsilon, T) \sim O(\frac{T^3}{N\epsilon^2})$. Otherwise, if G is unstable, $\Delta(\epsilon, T)$ grows exponentially with T. $\square$

Example 5. *Continuing with the system described in Example 4, the cost of privacy associated with the communication strategy of* (7.28) *is established. In this example,* $K = 0.2I$*. The coupling parameter* c *is chosen to be* 0.4*. Then, the closed-loop system is stable. Therefore, the sensitivity is bounded by* $\kappa(t) = 1.2 - 0.2 \times 0.6^t$*. The cost of privacy of the system with* N *agents at time* T *follows* $\frac{0.24(T-1)^3}{N\epsilon^2} + O(\frac{T^2}{N\epsilon^2})$*.*

Example 6. *Consider a linear distributed control system in which each agent is a point on the plane moving towards a randomly chosen destination with dynamics described in Example 5 and control strategies given in Example 5.*

The cost of each agent is defined by the distance between its position to its destination. The coupling between agents is the repulsive force in the direction of the center of gravity (CM) of the population. Thus, if the control of an individual fights the force too strongly without the knowledge of the CM then a higher cost is incurred. The system is numerically simulated with different levels of privacy and different distributions of destinations and make the following observations.

Figure 7.1 shows the relative costs of control with (dark blue) no communication and (light green) private communication, with respect to cost of control with complete (or broadcast) communication. First of all, if both the initial positions and the destinations are chosen with 0 mean, then the CM of the population hovers around the origin and in that case, the contribution of the coupling is small. As a result, there is not much to be gained through communication and the cost of the system with privacy is comparable to the cost of the system with no communication.

When the destination comes from some biased (nonzero mean) distributions, the cost of control with private communication starts to become smaller compared to those of systems with no communications.

Figure 7.2 shows that for the same distribution of initial positions and destinations the cost of privacy changes as predicted by Theorem 10. First of all, a higher level of privacy comes with a higher cost (Figure 7.2a). Secondly, a larger number of agents (N*) gives a lower cost of privacy (Figure 7.2b). As* N *changes from* 10 *to* 100*, the CoP decreases from* 4 *to* 0.4*. And finally a longer time horizon (*T*) translates to higher costs (Figure 7.2c). The simulation results matches the*

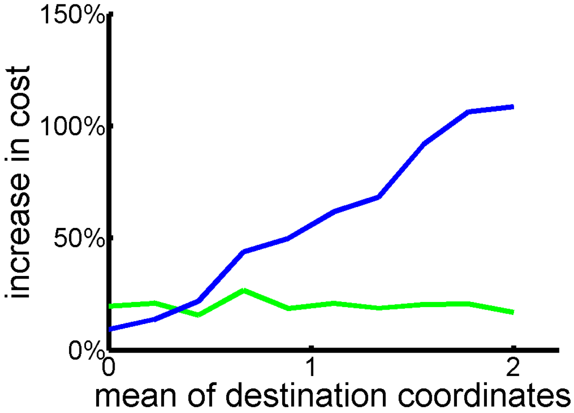
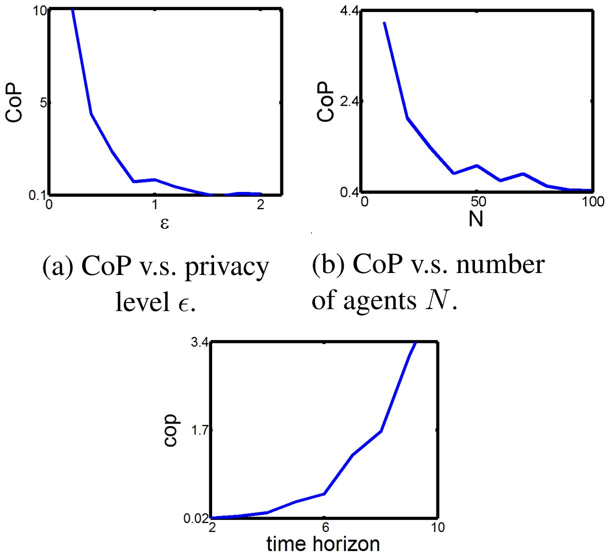


Figure 7.1: Increase in cost with biased sampled destinations. The blue and green lines capture the relative cost of control with no communication and private communication with respect to the cost of control with broadcast preferences respectively.

theoretical result that the cost of privacy has the order of $O(\frac{T^3}{N\epsilon^2})$.



(a) CoP v.s. privacy level ϵ.

(b) CoP v.s. number of agents N.

(c) CoP v.s. time horizon T.

Figure 7.2: Cost of Privacy for different privacy level, number of agents and time horizon.

CHAPTER 8

DIFFERENTIAL PRIVACY AND ENTROPY

In this section, the impact of differential privacy on estimating the private data of the system from the noisy communication is studied, namely the sequence of reported states $O_D = \{\tilde{x}(t)\}_{t<T}$. It can be shown that under certain technical assumptions, there is an optimal mechanism of adding correlated noise that minimizes the entropy of unbiased estimators on the private data.

First, it is assumed that the noise $n(t)$ is mean-zero, namely $\mathbb{E}[n(t)] = 0$. To facilitate further discussion, the dynamics of distributed system (7.11) (7.12) is written in the following aggregated form,

$$x(t+1) = (\mathbf{K} + \mathbf{C})x(t) - \mathbf{C}\tilde{x}(t) + (I - \mathbf{K})p(t+1), \tag{8.1}$$

$$\begin{aligned} x(t) =&(\mathbf{K} + \mathbf{C})^t x(0) \\ &+ \sum_{s=0}^{t-1} (\mathbf{K} + \mathbf{C})^{t-s-1}((I - \mathbf{K})p(s+1) - \mathbf{C}\tilde{x}(t)), \end{aligned} \tag{8.2}$$

where

$$\tilde{x}(t) = x(t) + n(t) \tag{8.3}$$

is the noisy observation, and

$$n(t) = (n_1(t), n_2(t), \ldots, n_N(t)) \in \mathcal{X}^N$$

is the *aggregated noise*.

Recall from Section 6.4 that $\hat{D}$ is an unbiased estimator of the private data D given observation O_D up to time $T - 1$. There is a lower bound on the entropy of the estimator.

Theorem 11. *If the private data $D = (x(0), p(1), \ldots, p(T-1))$ is ϵ-Differentially*

Private and $(I - K)$ is invertible, then the entropy of any unbiased estimator $\hat{D}$ of the private data satisfies

$$H(\hat{D}) \geq Nn(1 - \ln(\epsilon/2)) + N(T-1)H((I-K)v), \tag{8.4}$$

where $v \sim$ Lap$(1/\epsilon, n)$. The minimum is achieved by adding noise

$$n(0) = v(0), \tag{8.5}$$

and for $t \geq 1$,

$$n(t) = (\mathbf{K} + \mathbf{C})^t v(0) + \sum_{s=1}^{t} (\mathbf{K} + \mathbf{C})^{t-s}(I - \mathbf{K})v(s), \tag{8.6}$$

where $v(t) \sim$ Lap$(1/\epsilon, nN)$ are independent for $t = 0, \ldots, T-1$.

To prove Theorem 11, following condition for ϵ-Differential Privacy is needed for the system.

Proposition 1. *If the private data*

$$D = (x(0), p(1), \ldots, p(T-1))$$

is ϵ-Differentially Private, then the probability density function $f(\theta|D)$ of the estimator $\hat{D}$ satisfies

$$|\hat{m} \cdot \nabla f(\theta|D)| \leq \epsilon f(\theta|D) \tag{8.7}$$

almost everywhere, where $\hat{m}$ is an arbitrary unit vector.

Proof. By (6.13), for any $\theta, \theta' \in \mathcal{X}^N$ and $y \in \mathcal{X}^N$,

$$f(\theta|D) \leq e^{\epsilon\|\theta - \theta'\|_1} f(\theta'|D).$$

Thus,

$$\frac{f(\theta|D) - f(\theta'|D)}{\|\theta - \theta'\|_1} \leq \frac{e^{\epsilon\|\theta-\theta'\|_1} - 1}{\|\theta - \theta'\|_1} f(\theta'|D). \tag{8.8}$$

Letting $\theta \to \theta'$gives $|\frac{\theta - \theta'}{\|\theta-\theta'\|_1} \cdot \nabla f(\theta'|D)| \leq \epsilon f(\theta'|D)$, for θ' almost everywhere, abbreviated as a.e. Since θ can approach θ' in arbitrary direction, it is proved that

$|\hat{m} \cdot \nabla f(\theta'|D)| \leq \epsilon f(\theta'|D)$ for arbitrary unit vector $\hat{n}$ and θ' a.e. $\square$

8.1 One-shot case

Consider the case of $T = 1$ where $D = (x(0))$ and

$$\tilde{x}(0) = x(0) + n(0) \tag{8.9}$$

The initial state $x(0)$ is kept ϵ-Differentially Private by adding a mean-zero noise n which may depend on the value of $x(0)$. Let $f(y|x)$ be the probability density function of the estimator $\hat{x}(0)$ when $x(0) = x$. Denote $p(x, y) = f(y|x)$. It is assumed that $p(x, y)$ is absolutely continuous in both x and y. In this case, the following equivalent condition for ϵ-Differential Privacy can be derived.

To minimizes the entropy of the estimation $\hat{x}(0)$, the noise $n(0)$ should have the following symmetric properties.

Lemma 5. *The entropy of estimator $\hat{x}(0)$ is minimized when the probability distribution function $f(y|x) = p(x, y)$ of the observation $\tilde{x}(0)$ satisfies that for any $x \in \mathcal{X}^N$, $q(x, y) = p(x, y - x)$ is an even function in each component y_m of y.*

Proof. Without loss of generality, assume $m = 1$. Let

$$H_1^+(p) = \sup_{x \in \mathcal{X}^N} \int_{[x_1, \infty) \times \mathbb{R}^{n-1}} -p(x, y) \ln p(x, y) \mathrm{d}y, \tag{8.10}$$

$$H_1^-(p) = \sup_{x \in \mathcal{X}^N} \int_{(-\infty, x_1] \times \mathbb{R}^{n-1}} -p(x, y) \ln p(x, y) \mathrm{d}y. \tag{8.11}$$

Define

$$p'(x, y) = \begin{cases} p(x, y) & \begin{array}{c} \text{if } y_1 > x_1, H_1^+(p) \leq H_1^-(p) \\ \text{or } y_1 < x_1, H_1^+(p) > H_1^-(p), \end{array} \\ p(x, z) & \begin{array}{c} \text{if } y_1 > x_1, H_1^+(p) > H_1^-(p) \\ \text{or } y_1 < x_1, H_1^+(p) \leq H_1^-(p). \end{array} \end{cases} \tag{8.12}$$

where

$$\begin{cases} z_i = 2x_i - y_i, & i = 1, \\ z_i = y_i, & i = 2, 3, \ldots, n. \end{cases} \tag{8.13}$$

It is easy to check that $p'(x,y)$ satisfies the constraints in Problem 1 and $f'(y|x) = p'(x, y-x)$ is an even function in each component y_m of y. Finally, the proof is finished by noting that $H(p') = 2\min\{H_1^+(p), H_1^-(p)\} \leq H_1^+(p) + H_1^-(p) = H(p)$, where the equality holds iff $H_1^+(p) = H_1^-(p)$. □

Lemma 6. *The entropy of estimator $\hat{x}(0)$ is minimized when the probability distribution function $f(y|x) = p(x,y)$ of the observation $\tilde{x}(0)$ satisfies that for any $a \in \mathbb{R}$ and $m \in [n]$, fixing $x_1, \ldots, x_{m-1}, x_{m+1}, \ldots, x_n$ and $y_1, \ldots, y_{m-1}, y_{m+1}, \ldots, y_n$,*

$$\begin{aligned} p(x,y) =& p(x_1, \ldots, x_{m-1}, 2a - x_m, x_{m+1}, \ldots, x_n, \\ & y_1, \ldots, y_{m-1}, 2a - y_m, y_{m+1}, \ldots, y_n) \end{aligned} \tag{8.14}$$

Proof. Fix $x_1, \ldots, x_{m-1}, x_{m+1}, \ldots, x_n$ and $y_1, \ldots, y_{m-1}, y_{m+1}, \ldots, y_n$ and write $p(x,y)$ as $p(x_m, y_m)$ for simplicity. Without loss of generality, assume $m = 1$. Let $L^+ = \{x \in \mathcal{X}^N | x_1 \geq t\}$ and $L^- = \{x \in \mathcal{X}^N | x_1 < t\}$. Define

$$H^+(p) = \sup_{L^+} \int_{\mathcal{X}^N} -p(x,y) \ln p(x,y) \mathrm{d}y, \tag{8.15}$$

$$H^-(p) = \sup_{L^-} \int_{\mathcal{X}^N} -p(x,y) \ln p(x,y) \mathrm{d}y. \tag{8.16}$$

If $H^+(p) \leq H^-(p)$, let

$$p'(x_1, y_1) = \begin{cases} p(x_1, y_1), & x_1 \in L^+, \\ p(2a - x_1, 2a - y_1), & x_1 \in L^-, \end{cases} \tag{8.17}$$

otherwise, let

$$p'(x_1, y_1) = \begin{cases} p(2a - x_1, 2a - y_1), & x_1 \in L^+, \\ p(x_1, y_1), & x_1 \in L^-. \end{cases} \tag{8.18}$$

It is easy to check that $p'(x,y)$ satisfies the constraints in Problem 1 and $p'(x_1, y_1) = p'(2a - x_1, 2a - y_1)$. Finally, the proof is finished by noting that $H(p') = \min\{H^+(p), H^-(p)\} \leq \max\{H^+(p), H^-(p)\} = H(p)$, where the equality holds iff $H^+(p) = H^-(p)$. □

The above two propositions imply that the optimal noise added is independent of the original data.

Lemma 7. *The entropy of estimator $\hat{x}(0)$ is minimized when the noise $n(0)$ is independent of the value of the initial state $x(0)$.*

Proof. Consider $q(x, y) = p(x, y - x)$, which is the probability distribution function of the noise $n(0)$ for fixed x. Lemma 6 shows that for any $t \in \mathbb{R}$ and $m \in [n]$,

$$q(x_m, y_m) = q(2t - x_m, -y_m).$$

Lemma 5 implies that

$$q(2t - x_m, -y_m) = q(2t - x_m, y_m).$$

Therefore, the value of $q(x, y)$ is independent of each x_m. □

Lemma 7 implies that

$$p(x, y) = f(y - x), \tag{8.19}$$

where f is a even function in each x_i. When $nN = 1$, minimizing the entropy of $\hat{x}(0)$ is equivalent to solving the following problem

Problem 1 (Scalar Case)**.**

$$\begin{aligned} \textit{Minimize: } & H(f) = -\int_{[0,\infty)} f(x) \ln f(x) \mathrm{d}x, \\ \textit{subject to: } & f(x) \textit{ is absolutely continuous}, \\ & f(x) \geq 0, \\ & |f'(x)| \leq \epsilon f(x) \textit{ a.e.}, \\ & \int_{[0,\infty)} f(x) \mathrm{d}x = \frac{1}{2}. \end{aligned}$$

Problem 1 can be solved with the following results.

Lemma 8. *$f(x)$ is non-increasing if it solves Problem 1.*

Proof. Let $f(x)$ be a function that solves Problem 1 and

$$g(x) = \sup_{y \geq x} f(y).$$

Clearly, $g(x) \geq f(x)$ for $x \geq 0$. Suppose that $f(x)$ is not non-increasing, namely, for some $x^* > 0$, $g(x^*) > f(x^*)$. By the continuity of f, there exists a "largest" non-empty interval (a, b) containing x^*, on which $g(x) > f(x)$. Note that b is finite since $f(x) > 0$ and $\lim_{x\to\infty} f(x) = 0$. In addition, $g(b) = f(b)$. Let

$$d = \frac{1}{f(a)} \int_a^b f(x)\mathrm{d}x, \tag{8.20}$$

where $d \in [0, b - a)$.

There are two cases on the value of a. If $a > 0$, then $f(a) = g(a) = f(b) = g(b)$. Define

$$h(x) = \begin{cases} f(x), & x \in [0, a], \\ f(b), & x \in [a, a + d], \\ f(x + b - a - d), & x \in [a + d, \infty], \end{cases} \tag{8.21}$$

Otherwise, $a = 0$. Define

$$h(x) = \begin{cases} f(b), & x \in [0, d], \\ f(x + b - d), & x \in [d, \infty], \end{cases} \tag{8.22}$$

In both cases, $h(x)$ satisfies the constraints in Problem 1 and $H(h) < H(f)$. This is in contradiction with the assumption. □

Solution of Problem 1. Let $F(x) = \int_x^\infty f(y)\mathrm{d}y$ and note that $f(\infty) = 0$. By the definition of ϵ-Differential Privacy,

$$\begin{aligned} \epsilon F(x) \geq \int_x^\infty |f'(x)|\mathrm{d}y \geq |\int_x^\infty f'(x)\mathrm{d}y| \\ = |f(\infty) - f(x)| = f(x), \end{aligned} \tag{8.23}$$

where the equalities hold iff $f'(x) = -\epsilon f(x)$ for x a.e. In particular, $f(0) \leq \epsilon F(0) = \epsilon/2$.

Lemma 8 implies that $f'(y) \leq 0$ a.e., thus

$$\begin{aligned}
H(f) &= -\int_0^\infty f(x)\ln f(x)\mathrm{d}x \\
&= -\int_0^\infty f(x)\left(\ln f(0) + \int_0^x \frac{f'(y)}{f(y)}\mathrm{d}y\right)\mathrm{d}x \\
&= -\frac{\ln f(0)}{2} - \int_0^\infty \frac{f'(y)}{f(y)}\left(\int_x^\infty f(x)\mathrm{d}x\right)\mathrm{d}y \\
&= -\frac{\ln f(0)}{2} - \int_0^\infty \frac{f'(y)F(y)}{f(y)}\mathrm{d}y \\
&\geq -\frac{\ln f(0)}{2} - \int_0^\infty \frac{f'(y)}{\epsilon}\mathrm{d}y \\
&= \frac{f(0)}{\epsilon} - \frac{\ln f(0)}{2},
\end{aligned} \tag{8.24}$$

where the equality holds iff $f'(x) = -\epsilon f(x)$.

Since $f(0) \in (0, \epsilon/2]$, on which $\epsilon f(0) - \frac{1}{2}\ln f(0)$ is decreasing, $H(f) \geq (1 - \ln(\epsilon/2))/2$. Again, the equality holds if $f'(x) = -\epsilon f(x)$ a.e.

In sum, $H(f)$ achieves the minimum $(1 - \ln(\epsilon/2))/2$ at $f'(x) = -\epsilon f(x)$. Applying the conditions that $f(x) \geq 0$ and $\int_{[0,\infty)} f(x)\mathrm{d}x = 1/2$ gives the solution to Problem 1,

$$f(x) = \frac{\epsilon e^{-x\epsilon}}{2}. \tag{8.25}$$

□

When $nN \geq 2$, the goal is to solve the following problem

Problem 2 (Multi-dimensional Case)**.**

$$\begin{aligned}
\textit{Minimize: } & H(f) = -\int_{\mathcal{X}_+^N} f(x)\ln f(x)\mathrm{d}x, \\
\textit{subject to: } & f(x) \textit{ is absolutely continuous}, \\
& f(x) \geq 0, \\
& |\frac{\partial f(x)}{\partial x_i}| \leq \epsilon f(x), \forall i \in [nN] \textit{ a.e.}, \\
& \int_{\mathcal{X}_+^N} f(x)\mathrm{d}x = \frac{1}{2^{nN}}.
\end{aligned}$$

Solution of Problem 2. For each fixed $x_2, x_3, \ldots, x_n$, let

$$g_{x_2,x_3,\ldots,x_n}(x_1) = f(x_1, x_2, \ldots, x_n), \tag{8.26}$$

then $g_{x_2,x_3,\ldots,x_n}(x_1) \geq 0$, $|g'_{x_2,x_3,\ldots,x_n}(x_1)| \leq \epsilon g_{x_2,x_3,\ldots,x_n}(x_1)$ and

$$\begin{aligned} H(f) = -\int_{\mathbb{R}_+^{n-1}} \int_{[0,\infty)} & g_{x_2,x_3,\ldots,x_n}(x_1) \\ & \ln g_{x_2,x_3,\ldots,x_n}(x_1) \mathrm{d}x_1 \mathrm{d}x_2 \mathrm{d}x_3 \ldots \mathrm{d}x_n. \end{aligned} \tag{8.27}$$

To minimize H, it is required that

$$\begin{aligned} f(x_1, x_2, \ldots, x_n) &= g_{x_2,x_3,\ldots,x_n}(x_1) \\ &= e^{-\epsilon x_1} h(x_2, x_3, \ldots, x_n) \end{aligned} \tag{8.28}$$

where $h(x_2, x_3, \ldots, x_n)$ is some function of $x_2, x_3, \ldots, x_n$. Repeating the above argument shows that the minimum is achieved by

$$f(x_1, x_2, \ldots, x_n) = ke^{-\epsilon(x_1+x_2+\ldots+x_n)} \tag{8.29}$$

where k is some constant. Finally, $\int_{\mathcal{X}_+^N} f(x)\mathrm{d}x = \frac{1}{2^{nN}}$, implies that $k = (\frac{\epsilon}{2})^{nN}$. In this case, the lower bound is $H(f) = \frac{nN}{2}(1 - \ln(\epsilon/2))$. □

From the above results, Theorem 11 is generalized to the following proposition.

Proposition 2. *Given invertible $M \in \mathbb{R}^{n\times n}$ and a randomizing mechanism $\tilde{x} = Mx + w$ that protects the ϵ-Differential Privacy of the private data $x \in \mathbb{R}^n$ by adding mean-zero noise $w \in \mathbb{R}^n$, the entropy of any unbiased estimator $\hat{x}$ from observation $\tilde{x}$ satisfies*

$$H(\hat{x}) \geq H(Mv), \tag{8.30}$$

and the minimum is achieved by adding noise $n = Mv$ where $v \sim Lap(1/\epsilon, n)$. In particular, when $M = I$, thus

$$H(\hat{x}) \geq n(1 - \ln(\epsilon/2)). \tag{8.31}$$

Proof. First, the proposition holds for $M = I$. In general, since M is invertible, from

$$M^{-1}\tilde{x} = x + M^{-1}w$$

the minimal entropy of the estimator is achieved by $M^{-1}w \sim \text{Lap}(1/\epsilon, n)$, namely $w = Mv$ where $v \sim \text{Lap}(1/\epsilon, n)$, and the minimal entropy is $H(Mv)$. □

8.2 General case

Theorem 11 is proved for the general case of $T > 1$ below.

Proof of Theorem 11. For simplicity, define

$$m(t) = \begin{cases} n(0), & \text{if } t = 0 \\ n(t) - (\mathbf{K} + \mathbf{C})n(t-1), & \text{else} \end{cases}. \tag{8.32}$$

(8.1) (8.2) (8.3) implies that

$$\tilde{x}(0) = x(0) + m(0), \tag{8.33}$$

and for $t \geq 1$,

$$\tilde{x}(t) - \mathbf{K}\tilde{x}(t-1) = (I - \mathbf{K})p(t) + m(t). \tag{8.34}$$

Therefore, the privacy and estimation of $x(0), p(1), \dots, p(T)$ are independent. Noting that

$$I - \mathbf{K} = I_N \otimes (I - K)$$

is invertible, by Proposition 2, the entropy-minimizing mechanism that protects the ϵ-Differential Privacy of $D = (x(0), p(1), \dots, p(T))$ is given by

$$m(t) = \begin{cases} v(0), & \text{if } t = 0 \\ (I - \mathbf{K})v(t), & \text{else} \end{cases}, \tag{8.35}$$

namely,

$$n(0) = v(0), \tag{8.36}$$

and for $t \geq 1$,

$$n(t) = (\mathbf{K} + \mathbf{C})^t v(0) + \sum_{s=1}^{t} (\mathbf{K} + \mathbf{C})^{t-s} (I - \mathbf{K}) v(s), \tag{8.37}$$

where $v(t) \sim \text{Lap}(1/\epsilon, nN)$ are independent for $t = 0, \ldots, T-1$. The minimal entropy of any unbiased estimator $\hat{D}$ of the private data is

$$\begin{aligned} H(\hat{D}) &= \sum_{t=0}^{T-1} H(m(t)) \\ &= Nn(1 - \ln(\epsilon/2)) + N(T-1)H((I-K)v), \end{aligned} \tag{8.38}$$

where $v(t) \sim \text{Lap}(1/\epsilon, n)$. □

CHAPTER 9

CONCLUSION

The first part of this thesis focuses on the statistical verification of stochastic hybrid system. Model reduction techniques have been performed on both Discrete-Time and Continuous-Time Stochastic Hybrid Systems to reduce them to Discrete-Time Markov Chains and Continuous-Time Markov Chains, respectively; and statistical verification algorithms have been proposed to verify Linear Inequality LTL and Metric Interval Temporal Logic on these discrete probabilistic models. The advantage of stratified sampling in verifying Probabilistic Computation Tree Logic on Labeled Discrete-Time Markov Chains are also demonstrated; this method can potentially be extended to other statistical verification algorithms to reduce computational costs.

The second part focuses on the Differential Privacy in distributed systems. The formulation and the Differential Privacy of the systems are formally defined. It is shown that there is a trade-off between the Differential Privacy and the tracking performance of the systems. In addition, the trade-off between the Differential Privacy and the unbiased estimation of the private data is demonstrated, and an optimal algorithm to achieve the best trade-off is given.

REFERENCES

[1] X. Jin, J. V. Deshmukh, J. Kapinski, K. Ueda, and K. Butts, "Benchmarks for model transformations and conformance checking," in *1st International Workshop on Applied Verification for Continuous and Hybrid Systems (ARCH)*, 2014.

[2] I. Daniele, F. Alessandro, H. Marianne, B. Axel, and P. Maria, "A smart grid energy management problem for data-driven design with probabilistic reachability guarantees," in *4th International Workshop on Applied Verification of Continuous and Hybrid Systems*, 2017, pp. 2–19.

[3] R. R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: the next computing revolution," in *Proceedings of the 47th design automation conference*. ACM, 2010, pp. 731–736.

[4] B. Liu, D. Hsu, and P. S. Thiagarajan, "Probabilistic approximations of ODEs based bio-pathway dynamics," *Theoretical Computer Science*, vol. 412, no. 21, pp. 2188–2206, May 2011.

[5] B. Liu, A. Hagiescu, S. K. Palaniappan, B. Chattopadhyay, Z. Cui, W.-F. Wong, and P. S. Thiagarajan, "Approximate probabilistic analysis of biopathway dynamics," *Bioinformatics*, vol. 28, no. 11, pp. 1508–1516, June 2012.

[6] P. Zuliani, "Statistical model checking for biological applications," *STTT*, pp. 1–10, Aug. 2014.

[7] B. M. Gyori, B. Liu, S. Paul, R. Ramanathan, and P. Thiagarajan, "Approximate probabilistic verification of hybrid systems," in *Hybrid Systems Biology*. Springer, 2015, pp. 96–116.

[8] T. Henzinger, P. Kopke, A. Puri, and P. Varaiya, "What's decidable about hybrid automata?" *Journal of Computer and System Sciences*, vol. 57, no. 1, pp. 94–124, 1998.

[9] E. Clarke, A. Fehnker, Z. Han, B. Krogh, J. Ouaknine, O. Stursberg, and M. Theobald, "Abstraction and Counterexample-Guided Refinement in Model Checking of Hybrid Systems," *JFCS*, vol. 14, no. 4, pp. 583–604, 2003.

[10] R. Alur, T. Dang, and F. Ivancic, "Counter-Example Guided Predicate Abstraction of Hybrid Systems," in *TACAS 2003*, 2003, pp. 208–223.

[11] N. Roohi, P. Prabhakar, and M. Viswanathan, "HARE: A Hybrid Abstraction Refinement Engine for verifying non-linear hybrid automata," in *Proceedings of TACAS*, 2017, pp. 573–588.

[12] P. Tabuada and G. Pappas, "Linear time logic control of discrete-time linear systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 12, pp. 1862–1877, Dec. 2006.

[13] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 287–297, Feb. 2008.

[14] T. Wongpiromsarn, U. Topcu, and R. M. Murray, "Receding horizon control for temporal logic specifications," in *Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '10. New York, NY, USA: ACM, 2010, pp. 101–110.

[15] J. Liu, N. Ozay, U. Topcu, and R. M. Murray, "Synthesis of reactive switching protocols from temporal logic specifications," *IEEE Transactions on Automatic Control*, vol. 58, no. 7, pp. 1771–1785, 2013.

[16] R. Chadha and M. Viswanathan, "A Counterexample Guided Abstraction-Refinement Framework for Markov Decision Processes," *ACM Transactions on Computational Logic*, vol. 12, no. 1, pp. 1:1–1:49, 2010.

[17] I. Tkachev and A. Abate, "Formula-free finite abstractions for linear temporal verification of stochastic hybrid systems," in *Proceedings of the 16th international conference on Hybrid Systems: Computation and Control*. ACM, 2013, pp. 283–292.

[18] I. Tkachev, A. Mereacre, J.-P. Katoen, and A. Abate, "Quantitative automata-based controller synthesis for non-autonomous stochastic hybrid systems," in *Proceedings of the 16th international conference on Hybrid Systems: Computation and Control*. ACM, 2013, pp. 293–302.

[19] A. J. Chorin, O. H. Hald, and R. Kupferman, "Optimal prediction and the mori-zwanzig representation of irreversible processes," *Proceedings of the National Academy of Sciences*, vol. 97, no. 7, pp. 2968–2973, Mar. 2000.

[20] C. Beck, S. Lall, T. Liang, and M. West, "Model reduction, optimal prediction, and the mori-zwanzig representation of markov chains," in *CDC/CCC*, 2009, pp. 3282–3287.

[21] G. Pola, A. Girard, and P. Tabuada, "Approximately bisimilar symbolic models for nonlinear control systems," *Automatica*, vol. 44, no. 10, pp. 2508 – 2516, 2008.

[22] A. Girard, G. Pola, and P. Tabuada, “Approximately bisimilar symbolic models for incrementally stable switched systems,” *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 116–126, 2010.

[23] M. Zamani, G. Pola, M. Mazo, and P. Tabuada, “Symbolic models for nonlinear control systems without stability assumptions,” *IEEE Transactions on Automatic Control*, vol. 57, no. 7, pp. 1804–1809, 2012.

[24] M. Zamani, P. M. Esfahani, R. Majumdar, A. Abate, and J. Lygeros, “Symbolic control of stochastic systems via approximately bisimilar finite abstractions,” *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3135–3150, 2014.

[25] H. L. S. Younes and R. G. Simmons, “Statistical probabilistic model checking with a focus on time-bounded properties,” *Information and Computation*, vol. 204, no. 9, pp. 1368–1409, Sep. 2006.

[26] N. Roohi, Y. Wang, M. West, G. Dullerud, and M. Viswanathan, “Statistical verification of the toyota powertrain control verification benchmark (to appear.),” 2017.

[27] Y. Wang, N. Roohi, M. West, M. Viswanathan, and G. E. Dullerud, “Statistical verification of dynamical systems using set oriented methods,” in *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, ser. HSCC ’15. New York, NY, USA: ACM, 2015, pp. 169–178.

[28] Y. Wang, N. Roohi, M. West, M. Viswanathan, and G. E. Dullerud, “A mori-zwanzig and mitl based approach to statistical verification of continuous-time dynamical systems**the authors acknowledge support for this work from nsf cps grant 1329991.” *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 267–273, 2015.

[29] Y. Wang, N. Roohi, M. West, M. Viswanathan, and G. E. Dullerud, “Verifying continuous-time stochastic hybrid systems via mori-zwanzig model reduction,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 3012–3017.

[30] P. Zuliani, C. Baier, and E. M. Clarke, “Rare-event verification for stochastic hybrid systems,” in *Proceedings of the 15th ACM International Conference on Hybrid Systems: Computation and Control*, ser. HSCC ’12. New York, NY, USA: ACM, 2012, pp. 217–226.

[31] K. Sen, M. Viswanathan, and G. Agha, “On statistical model checking of stochastic systems,” in *Computer Aided Verification*, ser. Lecture Notes in Computer Science, K. Etessami and S. K. Rajamani, Eds. Springer Berlin Heidelberg, Jan. 2005, no. 3576, pp. 266–280.

[32] H. L. S. Younes, "Error control for probabilistic model checking," in *Verification, Model Checking, and Abstract Interpretation, 7th International Conference, VMCAI 2006, Charleston, SC, USA, January 8-10, 2006, Proceedings*, 2006, pp. 142–156.

[33] N. Roohi, Y. Wang, M. West, G. Dullerud, and M. Viswanathan, "Statistical verification of the Toyota powertrain control verification benchmark," in *Proceedings of HSCC*, 2017, pp. 65–70.

[34] H. L. S. Younes, "Ymer: A statistical model checker," in *Computer Aided Verification*, ser. Lecture Notes in Computer Science, K. Etessami and S. K. Rajamani, Eds., no. 3576. Springer Berlin Heidelberg, 2005, pp. 429–433.

[35] H. L. S. Younes and R. G. Simmons, "Statistical probabilistic model checking with a focus on time-bounded properties," *Information and Computation*, vol. 204, no. 9, pp. 1368–1409, 2006.

[36] K. Sen, M. Viswanathan, and G. Agha, "Statistical model checking of black-box probabilistic systems," in *Computer Aided Verification*, ser. Lecture Notes in Computer Science, R. Alur and D. A. Peled, Eds., no. 3114. Springer Berlin Heidelberg, 2004, pp. 202–215.

[37] K. Sen, M. Viswanathan, and G. Agha, "Vesta: A statistical model-checker and analyzer for probabilistic systems," in *Quantitative Evaluation of Systems, 2005. Second International Conference on the*, 2005, pp. 251–252.

[38] K. Sen, M. Viswanathan, and G. Agha, "On statistical model checking of stochastic systems," in *Computer Aided Verification*, ser. Lecture Notes in Computer Science, K. Etessami and S. K. Rajamani, Eds., no. 3576. Springer Berlin Heidelberg, 2005, pp. 266–280.

[39] K. G. Larsen and A. Legay, "Statistical model checking: Past, present, and future," in *Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques*. Springer, Cham, 2016, pp. 3–15.

[40] E. M. Clarke and P. Zuliani, "Statistical model checking for cyber-physical systems," in *Automated Technology for Verification and Analysis*. Springer, Berlin, Heidelberg, 2011, pp. 1–12.

[41] D. Henriques, J. G. Martins, P. Zuliani, A. Platzer, and E. M. Clarke, "Statistical model checking for markov decision processes," in *2012 Ninth International Conference on Quantitative Evaluation of Systems*, 2012, pp. 84–93.

[42] J. Liu, *Monte Carlo Strategies in Scientific Computing*. Springer, 2008.

[43] P. A. Maginnis, M. West, and G. E. Dullerud, "Variance-reduced simulation of lattice discrete-time markov chains with applications in reaction networks," *Journal of Computational Physics*, vol. 322, pp. 400–414, 2016.

[44] F. Koufogiannis, S. Han, and G. J. Pappas, "Computation of privacy-preserving prices in smart grids," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 2142–2147.

[45] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 901–914.

[46] M. Xue, W. Wang, and S. Roy, "Security concepts for the dynamics of autonomous vehicle networks," *Automatica*, vol. 50, no. 3, pp. 852–857, 2014.

[47] E. Shi, T.-H. H. Chan, E. G. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data." in *19th Annual Network & Distributed System Security Symposium (NDSS)*, vol. 2, no. 3, 2011, p. 4.

[48] Y. Zhang, Y. Zhang, and K. Ren, "Distributed privacy-preserving access control in sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 8, pp. 1427–1438, Aug 2012.

[49] C. Dwork, "Differential privacy," in *Automata, languages and programming*. Springer, 2006, pp. 1–12.

[50] C. Dwork, "Differential privacy: A survey of results," in *Theory and applications of models of computation*. Springer, 2008, pp. 1–19.

[51] C. Dwork, M. Naor, G. Rothblum, and T. Pitassi, "Differential privacy under continual observation," in *Proceedings of the 42nd ACM symposium on Theory of computing*, 2010.

[52] M. Hardt and K. Talwar, "On the geometry of differential privacy," in *Proceedings of the 42nd ACM symposium on Theory of computing*. ACM, 2010, pp. 705–714.

[53] Z. Huang, S. Mitra, and G. Dullerud, "Differentially private iterative synchronous consensus," in *Proceedings of the 2012 ACM workshop on Privacy in the electronic society*, ser. WPES '12. New York, NY, USA: ACM, 2012, pp. 81–90.

[54] J. Le Ny and G. J. Pappas, "Differentially private kalman filtering," in *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*. IEEE, 2012, pp. 1618–1625.

[55] M. Hale and M. Egerstedt, "Cloud-based optimization: A quasi-decentralized approach to multi-agent coordination," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, Dec 2014, pp. 6635–6640.

[56] M. Hale and M. Egerstedty, “Differentially private cloud-based multi-agent optimization with constraints,” in *American Control Conference (ACC), 2015*, July 2015, pp. 1235–1240.

[57] S. Han, U. Topcu, and G. Pappas, “Differentially private convex optimization with piecewise affine objectives,” in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, Dec 2014, pp. 2160–2166.

[58] Y. Mo and R. Murray, “Privacy preserving average consensus,” in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, Dec 2014, pp. 2154–2159.

[59] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” in *In Proceedings of TCC*, 2006.

[60] J. Le Ny, “On differentially private filtering for event streams,” in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*. IEEE, 2013, pp. 3481–3486.

[61] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi, “Broadening the scope of differential privacy using metrics,” in *Privacy Enhancing Technologies*. Springer, Berlin, Heidelberg, 2013, pp. 82–102.

[62] M. Hardt and K. Talwar, “On the geometry of differential privacy,” in *Proceedings of the 42nd ACM symposium on Theory of computing*, ser. STOC ’10. New York, NY, USA: ACM, 2010, pp. 705–714.

[63] J. Reed and B. C. Pierce, “Distance makes the types grow stronger: a calculus for differential privacy,” in *Proceedings of the 15th ACM SIGPLAN international conference on Functional programming*, ser. ICFP ’10. New York, NY, USA: ACM, 2010, pp. 157–168.

[64] F. McSherry and K. Talwar, “Mechanism design via differential privacy,” in *Foundations of Computer Science, 2007. FOCS ’07. 48th Annual IEEE Symposium on*, oct. 2007, pp. 94 –103.

[65] C. Li, M. Hay, V. Rastogi, G. Miklau, and A. McGregor, “Optimizing linear counting queries under differential privacy,” in *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, ser. PODS ’10. New York, NY, USA: ACM, 2010, pp. 123–134.

[66] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, “Our data, ourselves: Privacy via distributed noise generation,” in *Advances in Cryptology - EUROCRYPT 2006*, ser. Lecture Notes in Computer Science, S. Vaudenay, Ed. Springer Berlin Heidelberg, 2006, vol. 4004, pp. 486–503.

[67] J. Le Ny and G. Pappas, "Differentially private filtering," *Automatic Control, IEEE Transactions on*, vol. 59, no. 2, pp. 341–354, Feb 2014.

[68] Q. Geng, P. Kairouz, S. Oh, and P. Viswanath, "The staircase mechanism in differential privacy," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 9, no. 7, pp. 1176–1184, Oct 2015.

[69] Y. Wang, N. Roohi, M. West, M. Viswanathan, and G. E. Dullerud, "Statistical verification of pctl using stratified samples," in *IFAC Conference on Analysis and Design of Hybrid Systems*, 2018, p. to appear.

[70] Y. Wang, N. Roohi, M. West, M. Viswanathan, and G. E. Dullerud, "Verifying stochastic hybrid systems via mori-zwanzig model reduction," *IEEE Transactions on Automatic Control*, p. under review, 2018.

[71] Y. Wang, Z. Huang, S. Mitra, and G. Dullerud, "Entropy-minimizing mechanism for differential privacy of discrete-time linear feedback systems," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, Dec 2014, pp. 2130–2135.

[72] Y. Wang, M. T. Hale, M. Egerstedt, and G. E. Dullerud, "Differentially private objective functions in distributed cloud-based optimization." in *CDC*, 2016, pp. 3688–3694.

[73] Y. Wang, Z. Huang, S. Mitra, and G. E. Dullerud, "Differential privacy in linear distributed control systems: Entropy minimizing mechanisms and performance tradeoffs," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 118–130, 2017.

[74] Y. Wang, S. Mitra, and G. E. Dullerud, "Differential privacy and minimum-variance unbiased estimation in multi-agent control systems," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9521–9526, 2017.

[75] Z. Huang, Y. Wang, S. Mitra, and G. E. Dullerud, "On the cost of differential privacy in distributed control systems," in *Proceedings of the 3rd international conference on High confidence networked systems*. ACM, 2014, pp. 105–114.

[76] O. Maler and D. Nickovic, *Monitoring Temporal Properties of Continuous Signals*, 2004, pp. 152–166.

[77] J. V. Deshmukh, A. Donzé, S. Ghosh, X. Jin, G. Juniwal, and S. A. Seshia, *Robust Online Monitoring of Signal Temporal Logic*, 2015, pp. 55–70.

[78] A. Donzé, T. Ferrère, and O. Maler, *Efficient Robust Monitoring for STL*, 2013, pp. 264–279.

[79] P. Gastin and D. Oddoux, “Fast ltl to büchi automata translation,” in *Proceedings of the 13th International Conference on Computer Aided Verification*, ser. CAV ’01. London, UK, UK: Springer-Verlag, 2001, pp. 53–65.

[80] A. Duret-Lutz, “Ltl translation improvements in spot,” in *Proceedings of the Fifth International Conference on Verification and Evaluation of Computer and Communication Systems*, ser. VECoS’11. Swinton, UK, UK: British Computer Society, 2011, pp. 72–83.

[81] A. Duret-Lutz and D. Poitrenaud, “Spot: an extensible model checking library using transition-based generalized büchi automata,” in *IN PROC. OF MASCOTS’04*. IEEE Computer Society, 2004, pp. 76–83.

[82] R. Alur, T. Feder, and T. A. Henzinger, “The benefits of relaxing punctuality,” *J. ACM*, vol. 43, no. 1, pp. 116–146, 1996.

[83] R. Alur and D. L. Dill, “A theory of timed automata,” *Theor. Comput. Sci.*, vol. 126, no. 2, pp. 183–235, Apr. 1994.

[84] A. R. Teel, A. Subbaraman, and A. Sferlazza, “Stability analysis for stochastic hybrid systems: A survey,” *Automatica*, vol. 50, no. 10, pp. 2435 – 2456, 2014.

[85] A. R. Teel and J. P. Hespanha, “Stochastic hybrid systems: a modeling and stability theory tutorial,” in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. IEEE, 2015, pp. 3116–3136.

[86] A. R. Teel, *Recent Developments in Stability Theory for Stochastic Hybrid Inclusions*. Cham: Springer International Publishing, 2017, pp. 329–354.

[87] A. Subbaraman and A. R. Teel, “Robust global recurrence for a class of stochastic hybrid systems,” *Nonlinear Analysis: Hybrid Systems*, vol. 25, pp. 283 – 297, 2017.

[88] I. Karatzas and S. Shreve, *Brownian motion and stochastic calculus*. Springer Science & Business Media, 2012, vol. 113.

[89] D. Revuz and M. Yor, *Continuous martingales and Brownian motion*. Springer Science & Business Media, 2013, vol. 293.

[90] M. Dellnitz and O. Junge, “On the approximation of complicated dynamical behavior,” *SIAM Journal on Numerical Analysis*, vol. 36, no. 2, pp. 491–515, Jan. 1999.

[91] Y. Kwon and G. Agha, “Linear inequality ltl (iltl): A model checker for discrete time markov chains,” in *Formal Methods and Software Engineering*, ser. Lecture Notes in Computer Science, J. Davies, W. Schulte, and M. Barnett, Eds. Springer Berlin Heidelberg, 2004, vol. 3308, pp. 194–208.

[92] A. Wald, “Sequential tests of statistical hypotheses,” *The Annals of Mathematical Statistics*, vol. 16, no. 2, pp. pp. 117–186, 1945.

[93] Y. S. Chow and H. Robbins, “On the asymptotic theory of fixed-width sequential confidence intervals for the mean,” *The Annals of Mathematical Statistics*, vol. 36, no. 2, pp. 457–462, 04 1965.

[94] T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White, “Testing closeness of discrete distributions,” *J. ACM*, vol. 60, no. 1, pp. 4:1–4:25, Feb. 2013.

[95] A. Agresti and B. A. Coull, “Approximate is better than “exact” for interval estimation of binomial proportions,” *The American Statistician*, vol. 52, no. 2, pp. 119–126, 1998.

[96] T. Tony Cai, “One-sided confidence intervals in discrete distributions,” *Journal of Statistical Planning and Inference*, vol. 131, no. 1, pp. 63–88, 2005.

[97] “PRISM - Case Studies.”

[98] D. Knuth and A. Yao, “Algorithms and complexity: New directions and recent results.” Academic Press, 1976.

[99] A. Itai and M. Rodeh, “Symmetry breaking in distributed networks,” *Inf. Comput.*, vol. 88, no. 1, pp. 60–87, 1990.

[100] D. Ghosh and C. Knapp, “Estimation of traffic variables using a linear model of traffic flow,” *Transportation Research*, vol. 12, no. 6, pp. 395 – 402, 1978.