Foundations of Probabilistic Programming

(tentative title)

Edited by

Gilles Barthe, Joost-Pieter Katoen, and Alexandra Silva

Contents

Quantitative Analysis of Programs with Probabilities and Concentration of Measure Inequalities

Sriram Sankaranarayanan
University of Colorado, Boulder, USA

Abstract: The quantitative analysis of probabilistic programs answers queries involving the expected values of program variables and expressions involving them, as well as bounds on the probabilities of assertions. In this chapter, we will present the use of concentration of measure inequalities to reason about such bounds. First, we will briefly present and motivate standard concentration of measure inequalities. Next, we survey approaches to reason about quantitative properties using concentration of measure inequalities, illustrating these on numerous motivating examples. Finally, we discuss currently open challenges in this area for future work.

Keywords: Concentration of Measure, Uncertainty Propagation, Probabilistic Programming.

1 Introduction

In this chapter, we present the use of concentration of measure inequalities for the quantitative analysis of probabilistic programs. A variety of approaches have focused on qualitative properties that involve the almost-sure satisfaction of temporal formulas involving the behaviors of programs with special attention towards the analysis of almost sure termination, recurrence and persistence (McIver and Morgan (2004); Esparza et al. (2012); Bournez and Garnier (2005); Chakarov and Sankaranarayanan (2013); Fioriti and Hermanns (2015); Kaminski et al. (2016); Chakarov et al. (2016); Dimitrova et al. (2016); Chatterjee et al. (2017, 2018); McIver et al. (2018)). On the other hand, quantitative properties include reasoning about probabilities of assertions involving conditions over the program state, expectations in-

volving the program variables, and expected time to program termination (Kaminski et al. (2016); Chatterjee et al. (2018)).

An important difficulty of quantitative analysis is the need to integrate over a potentially large number of random variables generated in a typical run of a probabilistic program in order to calculate the quantity of interest. Often, these variables are manipulated using nonlinear functions over the course of long running loops that calculate the result of the program. Thus, the result is quite often a nonlinear function involving a large number of random variables. To make matters worse, the function is represented only indirectly as the computer program itself. Reasoning about such functions can be quite challenging and is normally performed in a case-by-case fashion, one program at a time, to ease the understanding of the behavior. Mechanizing this process to yield a more automated analysis approach can be quite challenging.

There are many approaches to tackle the challenge of quantitative reasoning over programs with probabilistic statements. One approach pioneered by McIver and Morgan annotates the program with assertions and *expectations* that serve the same role as loop invariants (Cf. McIver and Morgan (2004)). This approach effectively represents the distributions over the intermediate states encountered during the execution at a sufficient level of abstraction to establish the property of interest for the program as a whole. The approach has also been mechanized using ideas from loop invariant synthesis (Cf. Katoen et al. (2010)), and extended to programs with distributions over continuous state variables (Cf. Chakarov and Sankaranarayanan (2013); Fioriti and Hermanns (2015); Chatterjee et al. (2018)).

In this chapter, we survey a related approach that uses concentration of measure inequalities — a set of elegant mathematical ideas that characterize how functions of random variables deviate from their expected value. More importantly, these inequalities place upper bounds on the probabilities of deviations of a particular magnitude. Paradoxically, they avoid the need for expensive integration and thus, become quite effective when deviations over a large number of random variables are considered. Most well known inequalities such as the Chernoff-Hoeffding bounds, however, suffer a number of limitations that prevent them from being directly applicable to the analysis of probabilistic programs. They require *independence* of the random variables involved, work only for random variables over *bounded sets of support*, and finally, prove concentrations over *sums* rather than more general functions of random variables. We show in this chapter how these limitations can be partly overcome through a series of increasingly more sophisticated inequalities and the *tricks* involved in applying them to specific situations.

```
angles = [10, 60, 110, 160, 140, ...
          100, 60, 20, 10, 0]
x := TruncGaussian(0,0.05,-0.5,0.5)
y := TruncGaussian(0, 0.1,-0.5,0.5)
# run for 100 repetitions
for reps in range(0,100):
#iterate through angles
  for theta in angles:
    # Distance travelled variation
    d = Uniform(0.98,1.02)
    # Steering angle variation
    t = deg2rad(theta) * (1 + ...
        TruncGaussian(0,0.01,-0.05,0.05))
    # Move distance d with angle t
    x = x + d * cos(t)
    y = y + d * sin(t)
#Probability that we went too far?
assert(x >= 277)
```

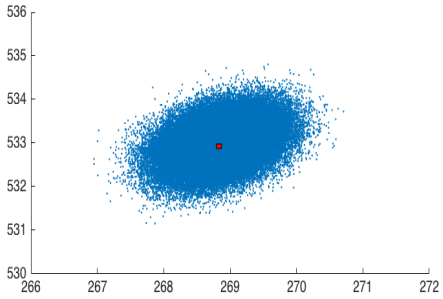


Figure 1 **Left**: A probabilistic program capturing the final position of 2D robotic end effector. **Right:** Scatter plot showing the final (x, y) values. Note that `TruncGaussian(m, s, l, u)` generates a truncated Gaussian random variable with mean m, standard deviation s, lower bound l and upper bound u.

The survey is based on previously published papers involving the author (see Chakarov and Sankaranarayanan (2013) and Bouissou et al. (2016)). We present concentration of measure inequalities motivated by a set of interesting numerical examples. We show applications to probabilistic programs starting with control deterministic computations that are handled through approximations known as probabilistic affine forms, whereas, more general loops are handled through the use of super martingale approaches. Our presentation is inspired by the excellent monograph on this topic by Dubhashi and Panconesi (2009). We recommend this book as a starting point towards more mathematically detailed presentations that include Williams (1991) and Boucheron et al. (2016).

1.1 Motivating Examples

In this section, we present motivating examples involving a robotic end effector, an anesthesia infusion process and a linear aircraft model under wind disturbances.

Example 1.1 (2D robotic end effector) Consider the repetitive motion of a 2D end effector used for tasks such as soldering printed circuit boards for manufacturing applications. The end effector makes a series of cyclic repetitive movements for each widget, ending each cycle at the starting position for soldering the subsequent widget. At each step, small calibration errors

can be introduced in its movement and these errors accumulate throughout the operation of the unit.

Figure 1 (left) shows the program that models the position of the end effector. Let (x, y) denote the position of the end effector. The initial position is defined by random variables (x_0, y_0) which are distributed as zero mean truncated Gaussian random variables over the set of support $[-0.5, 0.5]$ (see Fig. 1, lines 3, 4). The program itself runs a for loop in line 6 for $N = 100$ iterations that represent 100 different repetitions of the same sequence of actions by the robot. Each iteration j consists of a $k = 10$ different geometric transformations of the robot's position that result in a sequence of coordinates $(x_{0,j}, y_{0,j}) \ldots, (x_{k+1,j}, y_{k+1,j}))$, wherein,

$$(x_{i+1,j}, y_{i+1,j}) = (x_{i,j} + d_{i,j}\cos(\theta_{i,j}), y_{i,j} + d_{i,j}\sin(\theta_{i,j}))\,,$$

for $i = 1, \ldots, k$. Here $d_{i,j}$ is defined as a uniform random variable over $[0.98, 1.02]$. The mean values of $\theta_{i,j}$ are defined in degrees using the array `angles` in Fig. 1 (line 2), with the uncertainties modeled in line 13. The starting position for iteration $j + 1$ is the end position at iteration j.

$$(x_{0,j+1}, y_{0,j+1}) = (x_{k+1,j}, y_{k+1,j})\,.$$

We are interested in the probability that the value of $x_{N,k+1} \geq 277$ (line 18), for $N = 100$ and $k = 10$. The value of $x_{N,k+1}$ is shown for 10^5 different runs of the program in the scatter plot in Figure 1(right) and none of these simulations violate the assertion of interest. Thus, we seek an upper bound on the probability of violating this assertion of the form:

$$\mathbb{P}(x \geq 277) \leq ?\,.$$

The challenge lies in obtaining nontrivial bounds for this program given that (a) it involves nonlinear transformations of random variables and (b) roughly 2000 independent random variables are involved in $N = 100$ iterations.

Example 1.2 (Anesthesia Infusion Model) The anesthesia model consists of a four-chamber pharmacokinetic model of the anesthetic Fentanyl that is administered to a surgical patient using an infusion pump (see McClain and Hug (1980)). This model has been used as part of automated anesthesia delivery systems (see Shafer et al. (1988); Yousefi et al. (2017)). We model an erroneous infusion that results in varying amounts of anesthesia infused over time as a truncated Gaussian random noise. The state of the model at time t is a vector of concentrations of anesthesia in various "chambers" of the body:

$$\mathbf{x}(t) : (x_1(t), x_2(t), x_3(t), x_4(t))$$

The target state variable $x_4(t)$ measures the concentration of anesthesia in the blood plasma. Variable $u(t)$ denotes the rate of anesthesia infusion at time t, and is an input to the model.

At each step, the model evolves as

$$\mathbf{x}(t+1) = A\mathbf{x}(t) + Bu(t)(1 + w(t))$$

The matrices A, B are specified as follows:

$$A: \begin{bmatrix} 0.9012 & 0.0304 & 0.0031 & 0 \\ 0.0139 & 0.9857 & 0 & 0 \\ 0.0015 & 0 & 0.9985 & 0 \\ 0.0838 & 0.0014 & 0.0001 & 0.9117 \end{bmatrix} \quad B: \begin{pmatrix} 0.2676 \\ 0.002 \\ 0.0002 \\ 0.0012 \end{pmatrix}$$

The disturbance $w(t)$ is a truncated Gaussian variable over the range $[-0.4, 0.4]$ with mean 0 and standard deviation $\sigma = 0.08$. These model the error in the infused anesthesia rate as a percentage of the commanded rate $u(t)$. This rate $u(t)$ is specified as the following fixed set of infusion rates and times:

t(100 seconds)	$[0, 8]$	$[8, 14]$	$[14, 20]$	$[20, 26]$	$[26, 32]$	$[32, 38]$	$[38, 56]$
$u(t)(\mu$mol/s)	60	64	66	68	64	62	60

The control inputs in this example are chosen for illustrative purposes, and do not carry medical significance. The goal is to check the probability that the infusion errors result either in too much anesthesia $x_4(5600) \geq 300ng/mL$ potentially causing loss of breathing or too little anesthesia $x_4(5600) \leq 150ng/mL$ causing consciousness during surgery.

Example 1.3 (Fixed-Wing UAV Collision) Fixed wing small UAVs are quite prone to wind disturbances. Thus, it is important to predict if a collision is imminent using short term forecast models based on a series of positions and velocities of the system.

Auto-regressive moving average state-space (ARMAX) models are an important class of data-driven time series models that enable such forecasts to be obtained over short time periods (Brockwell and Davis (2009)). Figure 2 shows such a forecast model for a small fixed wing UAV inferred using ridge regression from data collected during test flights. The data reports GPS positions (x, y, z) and velocities (v_n, v_e, v_d), respectively, in the north, east and downward directions every $h = 0.18$ seconds for a period of 3 hours. Once the model is inferred, the residual errors between the model prediction and actual results are histogrammed. Often these residuals are modeled using Gaussian distributions with some statistical analysis. Here, we simply model them as unknown distributions whose means and standard deviations are given.

$$
\begin{array}{rll}
x(t+h) & = x(t) + hv_e(t) + e_x(t+h) & \\
y(t+h) & = y(t) + hv_n(t) + e_y(t+h) & \\
z(t+h) & = z(t) + hv_d(t) + e_z(t+h) & \\
v_n(t+h) & = 2.035v_n(t) - 1.11v_n(t-h) + 0.075v_n(t-2h) + w_1 & \leftarrow \sigma_1 : 0.055 \\
v_e(t+h) & = 1.923v_e(t) - 0.923v_e(t-h) + w_2 & \leftarrow \sigma_2 : 0.057 \\
v_d(t+h) & = 1.626v_d(t) - 0.778v_d(t-h) + 0.109v_d(t-2h) + w_3 & \leftarrow \sigma_3 : 0.16 \\
e_x(t+h) & = 0.567e_x(t) + 0.388e_x(t-h) + w_4 & \leftarrow \sigma_4 : 0.13 \\
e_y(t+h) & = 0.491e_y(t) + 0.27e_y(t-h) + 0.201e_y(t-2h) + w_5 & \leftarrow \sigma_5 : 0.14 \\
e_z(t+h) & = 1.35e_z(t) - 0.39e_z(t-h) + w_6 & \leftarrow \sigma_6 : 0.053
\end{array}
$$

Figure 2 Data-driven ARMAX model for predicting the future position of a UAV from its past positions and velocities. The time step h is 0.18 seconds in our model, x, y, z represent the position of the UAV, v_n, v_e, v_d represent the velocities in the north, east and downward directions, respectively, $e_x(t) : x(t) - x(t-h) - hv_e(t-h)$ is the deviation along the x direction, and similarly e_y, e_z denote deviations from y, z directions. $w_1, \ldots, w_6$ are residual errors that have been modeled using distributions with 0 mean and empirically estimated standard deviations σ_i shown alongside.

Using the model in Figure 2, we seek to build a *predictive monitor* that given the current history of positions, velocities and deviations

$$(x(t), x(t-h), y(t), y(t-h), \cdots, e_z(t), e_z(t-h)) \ ,$$

estimates a bound on the probability:

$$\mathbb{P}((x(t+Nh), y(t+Nh), z(t+Nh)) \in U) \leq ?$$

where U represents unsafe regions in the airspace denoted by proximity to buildings, grounds and designated no fly zones.

2 Quantitative Analysis: Problem and Approaches

In this section, we formally define the overall problem of quantitative analysis of probabilistic programs, focusing on (a) the type of systems that can be addressed, (b) the type of properties, and (c) sets of approaches that have been developed to reason about quantitative properties of probabilistic programs.

2.1 Programs and Properties

Given a "purely" probabilistic program P that computes a function $\mathbf{y} := F_P(X)$ over some random variables X, quantitative questions can be of two types: (a) bounds on the probability of an assertion φ involving $\mathbf{y}$:

$\mathbb{P}(\varphi(\mathbf{y})) \bowtie c$? and (b) bounds on the expectation of some function $g(\mathbf{y})$: $\mathbb{E}(g(\mathbf{y})) \bowtie c$? wherein $\bowtie \in \{\geq, \leq, =\}$ and c is a constant? Some of these questions are illustrated by our motivating examples from Section 1.1. As mentioned earlier, quantitative reasoning about the running time of programs is addressed elsewhere in this volume, although the approaches mentioned in this chapter remain generally applicable.

Beyond purely probabilistic programs, we may consider programs P that involve a combination of random variables X, *demonic* variables $\mathbf{w}$ controlled by the adversary, and angelic variables $\mathbf{u}$ controlled by a cooperative player. In such a situation, the program itself can be viewed as computing a joint function $\mathbf{y} := F_P(X, \mathbf{w}, \mathbf{u})$, wherein $\mathbf{y}$ denotes the outputs of the program. Interpreting $\varphi(\mathbf{y})$ as a *failure* condition, we wish to know if

$$(\exists\ \mathbf{u})\ (\forall\ \mathbf{w})\ \mathbb{P}_X(\varphi(\mathbf{y})) \leq c\,,$$

wherein c denotes a constant that is a desired failure threshold. We will focus our initial discussions on the case of purely probabilistic programs.

Furthermore, the probabilistic program will be assumed to be free of conditioning operation through `observe` or `assume` statements. Conditioning remains an open challenge for quantitative analysis and somewhat orthogonal to the purposes of quantitative reasoning considered in this chapter. Conditioning can simply be eliminated in restricted cases by computing the posterior distributions explicitly in the case of conjugate prior/posterior, or wherever symbolic integration approaches can tell us about the form of the posterior distribution (Narayanan et al. (2016); McElreath (2015)). Another approach involves the use of variational inference that can substitute prior probabilities by approximate posteriors from a predefined family of posterior distributions (Wingate and Weber (2013)).

Approaches to quantitative reasoning in probabilistic programs can be broadly classified into two: (a) simulation-based approaches and (b) symbolic approaches.

2.2 Simulation-Based Quantitative Reasoning

Simulation-based approaches execute the given program by sampling from the probability distributions generated in order to evaluate the property at hand. These approaches have been tied to statistical reasoning through hypothesis testing, starting with the work of Younes and Simmons (2006), leading to so-called *statistical model checking* approaches (Clarke et al. (2009); Agha and Palmskog (2018); Jha et al. (2009)).

Consider a probabilistic program P whose output variables are denoted

as $\mathbf{y}$ and a quantitative property $\mathbb{P}(\varphi(\mathbf{y})) \leq c$. A simulation based approach consists of two components: (a) generate samples $\mathbf{y}_1, \ldots, \mathbf{y}_N$ and (b) perform a statistical hypothesis test between two competing hypotheses:

$$\mathcal{H}_0 \ := \ \mathbb{P}(\varphi(\mathbf{y}) \leq c) \text{ versus } \mathcal{H}_1 \ := \ \mathbb{P}(\varphi(\mathbf{y}) > c)\,.$$

In particular, the hypothesis test works in a *sequential* fashion by examining how each added sample contributes towards the goal of accepting one hypothesis and rejecting another, with a new batch of samples generated *on-demand*.

To this end, the two most frequently used hypothesis tests include the sequential probability ratio test (SPRT) first proposed by Wald (1945) and the Bayes factor test proposed by Jeffries (Kass and Raftery (1995)). Details of these statistical tests are available from standard references, including the recent survey by Agha and Palmskog (2018). For instance, the Bayes factor test computes the so-called Bayes factor which is given by

$$\mathsf{BayesFactor} := \frac{\mathbb{P}(\text{Observations } \mathbf{y}_1, \ldots, \mathbf{y}_N \mid \mathcal{H}_1)\mathbb{P}(\mathcal{H}_1)}{\mathbb{P}(\text{Observations } \mathbf{y}_1, \ldots, \mathbf{y}_N \mid \mathcal{H}_0)\mathbb{P}(\mathcal{H}_0)}$$

as a measure of the evidence in favor of hypothesis $\mathcal{H}_1$ against that in favor of $\mathcal{H}_0$. Here, $\mathbb{P}(\mathcal{H}_j)$ refers to the prior probability of the hypothesis $\mathcal{H}_j$ for $j = 0, 1$. If the resulting BayesFactor exceeds a given upper bound threshold (see Kass and Raftery (1995) for an interpretation of the Bayes factor), the hypothesis $\mathcal{H}_1$ is accepted. On the other hand, if the BayesFactor falls below a lower bound, $\mathcal{H}_0$ is accepted in favor of $\mathcal{H}_1$. If the BayesFactor remains between these two bounds more evidence is sought since the data has insufficient evidence.

Besides the use of statistical tests, the generation of samples is another key problem. Often, in verification problems, the event of interest is a "rare" failure whose probability needs to be bounded by a small number $c \sim 10^{-6}$. To this end, the number of simulations needed can be prohibitively expensive, in practice. Thus, approaches such as importance sampling are used to artificially inflate the probability of obtaining a failure (see Srinivasan (2002); Bucklew (2004); Rubinstein and Kroese (2008)). Importance sampling approach first modifies the probabilistic program by replacing the distribution of random variables using sampling distributions designed to increase the probability and hence the number of samples that satisfy the assertion $\varphi(\mathbf{y})$ (assuming that φ is a rare event). The new samples are weighted by the ratio of the likelihood score under the original distribution and the new sampling distribution. A key challenge lies in designing a sampling distribution that can increase the number of rare event observations. This requires a lot of

insight on the part of the analyzer. Approaches such as the cross-entropy method can be employed to systematically optimize the parameters of a family of sampling distributions to make failures more likely (Jégourel et al. (2012); Sankaranarayanan and Fainekos (2012)).

2.3 Symbolic Approaches

In contrast to simulation-based approaches, symbolic techniques focus on reasoning about probabilities of assertions and expectations through a process of *abstraction*. Often this abstraction takes one of two forms (see Cousot and Monerau (2012) for a more refined classification): (a) *abstractions of intermediate probability distributions over program states* or (b) *abstractions of intermediate states as functions over the random variables generated by the program*. Both approaches rely on symbolic integration to compute bounds on the probabilities and expectations.

Abstractions of Probability Distributions: The probability distributions over program variables can be precisely represented for finite state programs. This is the basis for the tool PRISM, that handles probabilistic programs over finite state variables by compiling them into Markov chains or Markov decision processes, depending on whether demonic/angelic nondeterminism is present (Kwiatkowska et al. (2011)). These approaches can be extended to infinite state systems using the idea of a game-based abstraction that allows us to treat some of the probabilistic choices as non-deterministic but controlled by a different player (Cf. Parker et al. (2006)).

Abstractions for infinite state probabilistic systems are more complicated since the intermediate joint probability distributions between the program variables can be arbitrarily complicated (Kozen (1981)). A variety of approaches have been employed to abstract the intermediate distributions through probabilistic abstract domains that associate upper/lower bounds on measures associated with sets of states (Monniaux (2000, 2005); Cousot and Monerau (2012)). Whereas initial approaches focused on intervals and polyhedral sets annotated with bounds, it became clear that the probability bounds can often become too large to be useful or alternatively, the number of subdivisions of the state-space needed becomes too high to maintain a desired level of precision. An alternative approach by Bouissou et al. (2012) uses ideas from imprecise probabilities such as Dempster-Shafer structures (Dempster (1967); Shafer (1976)) and P-boxes(Ferson et al. (2003)) to represent probabilities more precisely. This approach has the added advantage of representing correlations between program variables in a more precise manner. However, the process of computing probabilities or expectations

involves integration, and therefore a summation over a large number of cells that tile the region of interest.

Probabilistic Symbolic Execution: A related and complementary approach uses symbolic execution to represent program states as functions over the input variables that involve random variables generated by the program (Geldenhuys et al. (2012); Mardziel et al. (2011); Sankaranarayanan et al. (2013)) followed by the use of symbolic integration to calculate the probability of an assertion exactly or approximately as needed. Algorithms for computing volume of polyhedra (De Loera et al. (2011)) or interval-based branch-and-bound schemes for approximating these volumes (Sankaranarayanan et al. (2013)) can be employed to perform quantitative analysis. A key drawback remains the high complexity of volume computation in terms of the number of dimensions of the region. Here, the dimensionality equals the number of random variables involved in the computation, which can be prohibitively large, as seen in our motivating examples. Thus, the applications are limited to programs that use fewer random variables and carry out complex computations over these. Furthermore, the exact volume computation is often not needed since for many applications of interest an upper bound over the probabilities of failure suffices.

3 Concentration of Measure Inequalities: A Primer

In this section, we present basic facts about concentration of measure inequalities. An accessible and complete exposition of concentration of measure inequalities and their application to randomized algorithms is available from Dubhashi and Panconesi (2009).

Concentration of measure inequalities allow us to reason about the behavior of certain functions of independent random variables. The most basic inequality remains the widely applied Chernoff-Hoeffding inequality. Let $X_1, \ldots, X_n$ be independent random variables taking on values in the set $\{0, 1\}$. Consider the sum $S_n = X_1 + \cdots + X_n$. Clearly, $\mathbb{E}(S_n) = \sum_{j=1}^{n} \mathbb{E}(X_j)$. The key question is how likely is it for the sum to satisfy $S_n \geq \mathbb{E}(S_n) + t$ for some positive deviation $t \geq 0$?

There are many ways of answering such a question. For the special case of $\{0, 1\}$-valued random variables that are identically distributed so that $\mathbb{E}(X_i) = p$ for all $i \in \{1, \ldots, n\}$, the answer can be obtained from an appli-

cation of combinatorics, as shown below:

$$\mathbb{P}\left(S_n \geq \mathbb{E}(S_n) + t\right) = \sum_{j=\lceil np+t \rceil}^{n} \binom{n}{j} p^j (1-p)^{n-j} \,.$$

The RHS expression provides an exact answer but is often cumbersome to compute. The expression can be approximated in many ways. For instance, the *Poisson approximation* is possible when n is large and p is small so that np is "small enough". However, such attempts produce a numerical approximation which cannot be used to establish guaranteed bounds, in general. Furthermore, we cannot deal with other common situations that involve: (a) the sum of random variables that are not necessarily identically distributed; (b) the sum of random variables whose distributions can be continuous; and finally (c) the sum of random variables that are not all independent.

Concentration of measure inequalities attempt to answer these questions by providing upper bounds on deviations of certain functions of random variables from their expected values. Let $f(X_1, \ldots, X_n)$ be a function of random variables having some fixed arity n (the arity of f does not need to be fixed, however). As an example: $f(X_1, \ldots, X_n) = X_1 + \cdots + X_n$. Let $\mathbb{E}(f)$ denote the expectation $\mathbb{E}(f(X_1, \ldots, X_n))$ computed over random choices of $X_1, \ldots, X_n$. A concentration of measure inequality typically has the form:

$$\mathbb{P}\left(f(X_1, \ldots, X_n) \geq \mathbb{E}(f) + t\right) \leq g(n, t) \,,$$

wherein $t \geq 0$, and g is a function that decreases sharply as t increases. Inequalities are often "symmetric" providing similar bounds for lower tails as well:

$$\mathbb{P}(f(X_1, \ldots, X_n) \leq \mathbb{E}(f) - t) \leq g(n, t) \,,$$

The inequality is *sub-gaussian* if the bound $g(n, t)$ is of the form $g(n, t) := C \exp\left(\frac{-ct^2}{n}\right)$ for known constants C, c that depend on the moments and set of support of the random variables $X_1, \ldots, X_n$. Most of the bounds we will explore will be sub-Gaussian in nature. The simplest and most fundamental of these bounds is the well-known Chernoff-Hoeffding inequality.

Theorem 1.4 (Chernoff-Hoeffding) *Let X_i be independent random variables that lies in the range $[a_i, b_i]$ almost surely, for $i = 1, \ldots, n$, and let $S_n = \sum_{j=1}^{n} X_j$. For all $t \geq 0$,*

$$\mathbb{P}(S_n \geq \mathbb{E}(S_n) + t) \leq \exp\left(-\frac{2t^2}{\sum_{j=1}^{n}(b_j - a_j)^2}\right) \,.$$

Using Chernoff-Hoeffding inequality, we may bound the upper tail of the sum of Bernoulli random variables as:

$$\mathbb{P}(S_n \geq \mathbb{E}(S_n) + t) \leq \exp\left(-\frac{2t^2}{n}\right).$$

However, there are two important limitations of Chernoff-Hoeffding inequality: (a) the random variables $X_1, \ldots, X_n$ must be independent and (b) X_i must lie within a bounded range $[a_i, b_i]$, almost surely.

Example 1.5 We will now illustrate the direct use of Chernoff-Hoeffding bounds to prove upper bounds on the probability of failure for the model described in Example 1.2. Note that our main object of concern in this example is the value of the state variable x_4 at time $t = 5600s$. Since at each step, the new state $\mathbf{x}(t+1)$ is related to the previous state: $\mathbf{x}(t+1) = A\mathbf{x}(t) + Bu(t)(1 + w(t))$, the value of $x_4(5600)$ is, in fact, written as a summation of the following form:

$$x_4(5600) = a_0 + \sum_{i=1}^{4} a_i x_i(0) + \sum_{j=1}^{5600} b_j w(j)\,, \tag{1}$$

wherein the coefficients a_i, b_j are obtained by computing the matrices for A^iB for $i = 0, \ldots, 5600$ and A^n for $n = 5600$. Furthermore, $w(j)$ for $j = 1, \ldots, 5600$, represent mutually independent random variables over the range $[-0.4, 0.4]$ with mean 0 and standard deviation $\sigma = 0.08$.

We may, therefore, apply Chernoff-Hoeffding bounds to compute bounds of the form:

$$\mathbb{P}(x_4(5600) \geq \mathbb{E}(x_4(5600)) + t) \leq \exp\left(\frac{-2t^2}{\sum_{i=1}^{5600} b_i (0.8)^2}\right),$$

and likewise,

$$\mathbb{P}(x_4(5600) \leq \mathbb{E}(x_4(5600)) - t) \leq \exp\left(\frac{-2t^2}{\sum_{i=1}^{5600} b_i (0.8)^2}\right).$$

Note that in applying these bounds, we consider the summation of random variables that include a_0, $a_i x_i(0)$ and $w(j)$ from Eq. (1). The value of $\mathbb{E}(x_4(5600))$ is calculated using linearity of expectation to be 246.7985 up to 4 significant digits. The denominator of the exponent term for the Chernoff-Hoeffding is calculated as follows:

$$\sum_{i=1}^{5600} b_i (0.8)^2 = 234.3159\,.$$

Thus, we may bound the probability of the Fentanyl concentration in the effect chamber exceeding 300ng/ml as follows:

$$\mathbb{P}(x_4(5600) \geq 300) \;\leq\; 3.05 \times 10^{-5}\,.$$

Likewise, we may bound the probability of Fentanyl concentration falling below 150ng/ml as follows:

$$\mathbb{P}(x_4(5600) \leq 150) \;\leq\; 2.4 \times 10^{-15}\,.$$

Chernoff-Hoeffding inequalities are widely used in numerous applications to the analysis of randomized algorithms for bounding away the probability of an undesirable behavior of the algorithm at hand. However, their use is constrained by many important factors:

(i) The inequality applies to random variables X_i whose set of support is bounded by a finite interval. Random variables with an unbounded set of support such as Gaussian random variables are not handled.
(ii) The inequality uses only the range and first moment of each X_i. Further information such as the second or higher moments $\mathbb{E}(X_i^2)$ could be more useful in obtaining sharper bounds.
(iii) The inequality applies to sums of random variables. Programs often compute more complex functions of random variables than just sums.
(iv) The inequality applies to mutually independent random variables. Even if the random variables sampled by a program are mutually independent, the state variables become correlated as they depend on the same set of independent random variables.

We will now discuss how each of the limitations may be handled using other, more sophisticated concentration of measure inequalities and/or simply by adapting how the inequality is applied in the first place.

3.1 Inequalities Using Higher Moments

Numerous inequalities for the concentration of the sum $\sum_{i=1}^{n} X_i$ of independent random variables have been proposed that use higher moments such as the second moment $\mathbb{E}(X_i^2)$ of each random variable X_i in addition to $\mathbb{E}(X_i)$. Bernstein (1924) proposed a series of such inequalities.

Theorem 1.6 (Bernstein Inequality) *Let $X_1, \ldots, X_n$ be independent random variables such that (a) there exists a constant $M > 0$ such that $|X_i - \mathbb{E}(X_i)| \leq M$ for each $i \in [1, n]$, and (b) the variance of each X_i is σ_i^2. For*

any $t \geq 0$*:*

$$\mathbb{P}(X - \mathbb{E}(X) \geq t) \;\leq\; exp\left(\frac{-t^2}{\frac{2}{3}Mt + 2\sum_{i=1}^{n}\sigma_i^2}\right) \tag{2}$$

For the left tail probability, we may derive an identical bound.

Note that if each random variable X_i ranges over a bounded interval, then condition (a) for Bernstein inequality is easily satisfied. Furthermore, if we let μ_i denote $\mathbb{E}(X_i)$, it is easy to show that if the interval $[\mu_i - \sigma_i, \mu_i + \sigma_i]$ for each random variable is small in comparison to the set of support $[a_i, b_i]$, this inequality will provide much tighter bounds when compared to Chernoff-Hoeffding bounds.

Example 1.7 Returning back to the analysis of the anesthesia model from Example 1.5, we will now apply Bernstein inequality to bound the probability that $x_4(5600) \geq 300$ng/ml. We can compute the sum of the variances $\sum_{i=1}^{5600}\sigma_i^2$ as 4.687. Similarly the value of M for Bernstein's inequality in (2) is calculated to be 2.362. These calculations are mechanized using the approach described in Section 4. Applying the inequality yields the bound:

$$\mathbb{P}(x_4(5600) \geq 300) \leq 7.1 \times 10^{-13}\,.$$

This is much more useful than the bound of 3.05×10^{-5} obtained using Chernoff-Hoeffding bounds. Similarly, the probability that the anesthesia level falls below the lower limit of 150ng/ml using Bernstein's inequality is obtained as 2.1×10^{-26}, once again a drastic improvement when compared to Chernoff-Hoeffding bounds.

Inequalities that use information from higher order moments beyond just the mean and the variance are also possible. In fact, these inequalities may be derived by using an expansion of the moment generating function $\mathbb{E}(e^{tX})$ for a random variable X whose set of support is bounded by $[a, b]$. The key lies in discovering useful bounds that can utilize as much information available about the random variables X_i as possible, while remaining computationally tractable. We see the use of such *designer inequalities* derived using computer algebra manipulations rather than using hand calculations as an important future step in mechanizing the application of concentration of measure inequalities.

3.2 Random Variables with Unbounded Support

All concentration of measure inequalities studied thus far, such as Chernoff-Hoeffding or Bernstein inequalities, rely on the random variables X_i having

bounded set of support. However, this need not be the case for many commonly encountered distributions such as Gaussian or exponential random variables.

Let $X_1, \ldots, X_n$ be independent random variables whose support is unbounded (either $[-\infty, \infty]$, $[a, \infty)$ or $(-\infty, a]$, for some constant a). We say that a family of distributions is *Lévy stable* iff the linear combination of finitely many random variables belonging to the family, is also a random variable that belongs to the family. For instance, commonly occurring distributions such as Gaussian, exponential, gamma, and Poisson are Lévy stable. If the variables X_i are identically distributed and their distributions are Lévy stable, then it is possible to calculate the parameters for the distribution of the sum from the parameters of the original random variables. Likewise, questions such as $\mathbb{P}(X \geq \mathbb{E}(X) + t)$ can be handled by knowing the cumulative density functions of these variables.

However, appealing to stability property of the random variables will fail if the distributions are not stable or, more commonly, the variables $X_1, \ldots, X_n$ are not identically distributed. In this situation, a simple trick can enable us to successfully apply concentration of measure inequality as follows:

(i) For each X_i choose an interval $J_i := [a_i, b_i]$ and compute the probability p_i that $\mathbb{P}(X_i \notin J_i)$ (or compute an interval bounding p_i). Also define a random variable Y_i obtained by restricting the variable X_i to the interval J_i. Let $\mathbb{E}(Y_i)$ be its expectation.

(ii) To bound the probability that $\mathbb{P}(\sum X_i \geq t)$, we can consider two mutually exclusive events. $A := \bigwedge X_i \in J_i$ and $B := \bigvee X_i \notin J_i$. We have that

$$\begin{aligned}
\mathbb{P}(\textstyle\sum X_i \geq t) &= \mathbb{P}(A)\mathbb{P}(\textstyle\sum X_i \geq t \,|\, A) + \mathbb{P}(B)\mathbb{P}(\textstyle\sum X_i \geq t \,|\, B) \\
&= \mathbb{P}(A)\mathbb{P}(\textstyle\sum Y_i \geq t) + \mathbb{P}(B)\mathbb{P}(\textstyle\sum X_i \geq t \,|\, B) \\
&\leq \mathbb{P}(A)\mathbb{P}(\textstyle\sum Y_i \geq t) + \mathbb{P}(B) \\
&\leq (\textstyle\prod_{i=1}^{n}(1-p_i))\mathbb{P}(\sum Y_i \geq t) + (1 - \prod_{i=1}^{n}(1-p_i))
\end{aligned}$$

Note that we obtain $\mathbb{P}(A) = \prod_{i=1}^{n}(1-p_i)$ through the independence of the random variables $X_1, \ldots, X_n$, and $\mathbb{P}(B) = 1 - \mathbb{P}(A)$. If independence of $X_1, \ldots, X_n$ is dropped (as we will see subsequently), we may instead use Fréchet bounds to conclude that $\mathbb{P}(A) \leq \min(1-p_1, \ldots, 1-p_n)$. Likewise, we may use a weaker bound $\mathbb{P}(B) \leq p_1 + \cdots + p_n$ through Boole's inequality (union bound) if the independence assumption is dropped. We may now estimate the probability $\mathbb{P}(\sum Y_i \geq t)$ using the Chernoff-Hoeffding bounds or Bernstein inequality (if the variance of Y_i is known).

The approach also presents an interesting trade-off between the size of

the interval J_i chosen for each random variable. A larger interval makes the probability $\mathbb{P}(B)$ vanishingly small. However, at the same time, the quality of the bounds depend on the width of the intervals J_i. For instance, the problem can be setup as an optimization to find the best bound that can be obtained by varying the width of J_i against the probability of event B.

Example 1.8 Returning to the anesthesia example (Ex. 1.2), we will consider the distribution of the noise to be a Gaussian random variable with mean 0 and variance 0.08. As a result, the concentration of measure inequalities are no longer applicable. However, if we consider $J_i := [-0.4, 0.4]$, we can estimate the probability $\mathbb{P}(w_i \notin J_i) \leq 5.73 \times 10^{-7}$. The latter is obtained knowing the probability that the value of a normally distributed random variables lies $\pm 5\sigma$ away from the mean. As a result, the result from the Chernoff-Bounds in Example 1.5 can be reused here to assert that

$$\mathbb{P}(x_4(5600) \geq 300) \leq \underbrace{(1 - 5.73 \times 10^{-7})3.05 \times 10^{-5} + 5600 \times 5.73 \times 10^{-7}}_{=3.293 \times 10^{-3}}$$

On the other hand, We could use a larger interval $J_i := [-0.593, 0.593]$ that yields the probability $\mathbb{P}(w_i \notin J_i) \leq 10^{-13}$. However, using this interval to truncate the random variable yields poorer results overall.

$$\mathbb{P}(x_4(5600) \geq 300) \leq 0.0012 + 5600 \times 10^{-13} \leq 0.0013\,.$$

The approach can also be used alongside Bernstein bounds provided the variance can be estimated for the truncated distribution. Here, we may use a formula for the variance of a truncated Gaussian distribution. In doing so with the larger interval $J_i := [-0.593, 0.593]$ we obtain a tighter bound:

$$\mathbb{P}(x_4(5600) \geq 300) \leq 3.241 \times 10^{-8} + 5600 \times 10^{-13} \leq 3.25 \times 10^{-8}\,.$$

3.3 Inequalities for Nonlinear Functions

Thus far, we have applied Chernoff-Hoeffding and Bernstein bounds for sums of independent random variables. However, more often, probabilistic programs yield nonlinear functions of random variables $f(X_1, \ldots, X_n)$. We are interested in tail bounds of the form

$$\mathbb{P}(f - \mathbb{E}(f) \geq t) \leq \exp(-ct^2)\,.$$

First, it is clear that not all functions will yield such a bound. It is important to understand properties of functions that are amenable to such a bound and check if the function computed by the program falls within such a class.

Example 1.9 Revisiting the 2D robotic end effector from Example 1.1, we note that the value of x at the end of the program in line 18 of Figure 1, is obtained as

$$x \;:=\; x_0 + \sum_{i=0}^{99}\sum_{j=0}^{9} d_{i,j}\cos(\theta_{i,j})\,. \tag{3}$$

wherein x_0 is a truncated Gaussian random variable with mean 0 and standard deviation 0.05 over the range $[-0.5, 0.5]$ (see line 3), $d_{i,j}$ is a uniform random variable over the range $[0.98, 1.02]$ and $\theta_{i,j}$ is given by

$$\theta_{i,j} = \alpha_j(1 + w_{i,j})$$

wherein α_j is specified in the array `angles` in line 2 of the program shown in Figure 1 and $w_{i,j}$ is distributed as a truncated Gaussian random variable with mean 0, standard deviation $\sigma = 0.01$ and over the range $[-0.05, 0.05]$ (see line 13).

Definition 1.10 (Difference Bounded Functions) Let $f(x_1, \ldots, x_n)$ be a function from $S_1 \times \cdots \times S_n \to \mathbb{R}$ for sets $S_i \subseteq \mathbb{R}$. We say that f is *difference bounded* iff there exists constants $c_1, \ldots, c_n$ such that

$$\begin{array}{l}(\forall i \in \{1, 2, \cdots, n\}) \\ \quad (\forall\ x_1 \in S_1, \ldots, x_{i-1} \in S_{i-1}, x_{i+1} \in S_{i+1}, \ldots, x_n \in S_n) \\ \quad\quad (\forall x_i \in S_i, x_i' \in S_i) \\ \quad\quad\quad |f(x_1, \ldots, \mathbf{x_i}, \ldots, x_n) - f(x_1, \ldots, \mathbf{x_i'}, \ldots, x_n)| \leq c_i.\end{array}$$

In other words, varying just the i^{th} argument while keeping the other arguments the same yields a bounded change in the value of the function. Dubhashi and Panconesi (2009) and many other authors sometimes use the terminology *Lipschitz* functions to refer to difference bounded functions, above. Note that the notion of difference bounded is not the same as the standard notion of Lipschitz continuity that one encounters in calculus, wherein the right hand side of the inequality is $L|x_i - x_i'|$ rather than a fixed constant c_i. It is easy to see that a Lipschitz continuous function is difference bounded provided the sets $S_1, \ldots, S_n$ are compact. On the other hand, the step function (a discontinuous function) is difference bounded over $[-1, 1]$ but not Lipschitz continuous.

A well known result called McDiarmid's inequality (McDiarmid (1989)) shows that a difference bounded function of independent random variables concentrates around its mean.

Theorem 1.11 (McDiarmid's Inequality) *Let $X_1, \ldots, X_n$ be independent*

random variables and f be a difference bounded function over the Cartesian product of the set of support of the random variables. We conclude that

$$\mathbb{P}(f(X_1, \ldots, X_n) \leq \mathbb{E}(f) + t) \leq \exp\left(\frac{-2t^2}{\sum_{j=1}^{n} c_j^2}\right).$$

A similar inequality holds for the lower tail, as well.

Example 1.12 Continuing with the calculation for Example 1.9, we will first show that the function in Equation (3) is difference bounded and derive the corresponding constants bounding the differences by hand:

Random Variable	Difference Bound Constant
x_0	1
$d_{i,j}$	$0.04u_j$
$w_{i,j}$	$1.02(u_j - l_j)$

Here

$$u_j \;:=\; \max(|\cos(0.95\alpha_j)|, |\cos(1.05\alpha_j)|)$$

and

$$l_j \;:=\; \min(|\cos(0.95\alpha_j)|, |\cos(1.05\alpha_j)|)\,.$$

Carrying out this calculation, the sum of the square of the difference bound constants is obtained as 13.68. Next we need to estimate $\mathbb{E}(f)$, which is challenging as it involves integrating a multivariate nonlinear function over the random variables. A systematic approach to doing so using a combination of affine forms, interval arithmetic and Taylor series expansions is described in our previous work (Bouissou et al. (2016)). Using an implementation of our approach, we estimate an interval that bounds the value of $\mathbb{E}(x)$ as

$$\mathbb{E}(x) \in [268.6170484, 270.6914916]\,.$$

Such a range is nevertheless useful in estimating tail probabilities. For instance, to bound upper tail probabilities $\mathbb{P}(f - \mathbb{E}(f) \geq t)$, we use the upper limit of the given range for $\mathbb{E}(f)$. Likewise, we use the the lower limit for the lower tail probabilities in order to obtain conservative bounds. Therefore, we conclude that

$$\mathbb{P}(x \geq 277) = \mathbb{P}(x - 270.69 \geq 6.31) \leq \exp\left(\frac{-2 * 6.31^2}{13.68}\right) = 2.96 \times 10^{-3}\,.$$

This bound is much improved using the systematic approach that incorporates variance information originally described in Bouissou et al. (2016), as will be discussed in the subsequent section.

3.4 Inequalities for Correlated Random Variables

We will now examine how concentration inequalities can be derived for dependent random variables $X_1, \ldots, X_n$. If the variables are correlated in some manner, it is hard to provide useful concentration bounds for the general case. However, in some cases, the "structure" of the correlation can be exploited to directly derive inequalities by adapting existing approaches such as Chernoff-Hoeffding or Bernstein inequalities.

Numerous cases have been studied such as *negatively dependent* random variables (Dubhashi and Panconesi (2009); Dubhashi and Ranjan (1998)). We will focus our approach on sums of random variables with a given *correlation graph*. Let $X_1, \ldots, X_n$ be a set of random variables with an undirected graph $G := (\{X_1, \ldots, X_n\}, E)$ whose vertices correspond to the random variables $X_1, \ldots, X_n$. An edge between two random variables (X_i, X_j) signifies a dependency between the variables.

Example 1.13 Let X_1, X_2 and X_3 be three independent random variables and X_4 denote a function $f(X_1, X_2, X_3)$. The dependency graph has edges connecting X_4 with X_1, X_2 and X_3.

Naturally, existing approaches discussed thus far require the random variables to be independent. As a result, it is not possible to apply them in this context. We will describe an elegant "trick" due to Janson (2004), and in turn following ideas from Hoeffding's seminal paper (Hoeffding (1963)) introducing the Chernoff-Hoeffding inequality.

First, we will introduce the notion of a weighted independent-set cover. Let A be the set of random variables $\{X_1, \ldots, X_n\}$. A subset $A_j \subseteq A$ is an independent set if any two variables in A_j are mutually independent, i.e, there are no edges between them in the graph G.

An *independent set cover* is a family of independent sets $A_1, \ldots, A_k$ such that $A_1 \cup \cdots \cup A_k = A$. A *weighted cover* is a family of independent sets with positive real-valued weights

$$(A_1, w_1), \ldots, (A_k, w_k)\,,$$

such that (a) $A_1, \ldots, A_k$ form an independent set cover and (b) for each X_i, $\sum_{A_j \mid X_i \in A_j} w_j \geq 1$. In other words, for each element X_i, the sum of weights for all independent sets that contain X_i is greater than or equal to 1. Note that every independent set cover that partitions the set A is also a weighted cover by assigning the weights 1 to each set. The total weight of a cover is given by $w_1 + \cdots + w_k$. Given a graph G its chromatic number $\xi(G) = k$, for some $k \in \mathbb{N}$, is the smallest number of sets that form an independent set

cover of A. Likewise, its *fractional chromatic number* $\xi^*(G)$ is the minimum weight $\sum_{j=1}^{k} w_j$ of some $A_1, \ldots, A_k$ such that $(A_1, w_1), \ldots, (A_k, w_k)$ forms a weighed cover.

Let $(A_1, w_1), \ldots, (A_k, w_k)$ be a weighted cover of the set of random variables A. Let $[a_i, b_i]$ represent the set of support for random variable X_i. Let $c_j := \sum_{X_i \in A_j} (b_i - a_i)^2$.

Theorem 1.14 (Janson (2004)) *Given a set of random variables $A := \{X_1, \ldots, X_n\}$ with correlations specified by graph G. Let $(A_1, w_1), \ldots, (A_j, w_j)$ be a weighted independent set cover of G. The following bound holds:*

$$\mathbb{P}(\sum X_j - \mathbb{E}(\sum X_j) \geq t) \leq \exp\left(\frac{-2t^2}{T^2}\right), \tag{4}$$

wherein $T^2 = \left(\sum_{j=1}^{k} w_j \sqrt{c_j}\right)^2$ and $c_j = \sum_{X_i \in A_j} (b_i - a_i)^2$.

With $\xi^*(G)$ as the fractional chromatic number of G, we obtain the bound

$$\mathbb{P}(\sum X_j - \mathbb{E}(\sum X_j) \geq t) \leq \exp\left(\frac{-2t^2}{\xi^*(G) \sum_{j=1}^{n} (b_j - a_j)^2}\right). \tag{5}$$

First we note that if all the variables are mutually independent, then the optimal weighted cover is simply $(A, 1)$ yielding $\xi^*(G) = 1$. Both Equations (4) and (5) yield the same answer as Chernoff-Hoeffding bounds. Applying the bound in (4) requires us to compute a weighted independent set cover of the graph G. A simple approach lies in using a greedy algorithm to partition the set A into subsets of independent sets, and using weights 1 to convert the cover into a weighted cover.

Example 1.15 Continuing with Example 1.13, an independent set cover is given by $\{X_1, X_2, X_3\}$ and $\{X_4\}$ which yields a weighted cover by assigning a weight 1 to each independent set.

Therefore, let $S := X_1 + X_2 + X_3 + X_4$ and $[a_i, b_i]$ denote the range of each random variable X_i. Applying Janson's inequality for any $t \geq 0$, we get:

$$\mathbb{P}(S \geq \mathbb{E}(S) + t) \leq \exp\left(\frac{-2t^2}{\left(\sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2 + (b_3 - a_3)^2} + (b_4 - a_4)\right)^2}\right).$$

Beyond Chernoff-Hoeffding bounds, Janson presents extensions of other inequalities such as Bernstein's inequality to the case of correlated random variables with known correlation structure.

Thus far, we have studied various concentration of measure inequalities and how they can be applied to reason about the probability of assertions for some specific programs. The bigger question, however, is to what extent can the process of choosing and applying the right inequality be mechanized for a given probabilistic program. To answer it, we examine the case of *control deterministic* programs and use the idea of affine forms to symbolically reason about the distribution of program variables during and after the program execution. This provides us a means to apply the inequalities we have discussed thus far in this section without requiring extensive manual calculations.

4 Control Deterministic Computations

In this section, we briefly touch upon how the concentration of measure inequalities presented in the previous sections can be systematically applied to reasoning about programs. We begin our discussion with a simple class of *control deterministic* computations. The material in this section is based upon joint work with Olivier Bouissou, Eric Goubault and Sylvie Putot (Cf. Bouissou et al. (2016)). Control determinism is an important property that is satisfied by many probabilistic programs that occur naturally in application domains such as cyber-physical systems (CPS), control theory, and motion planning, to name a few. In this section, we briefly summarize the notion of control determinism and examine how probability distributions of variables can be abstracted in a symbolic fashion, to enable reasoning using various concentration of measure inequalities.

4.1 Control Deterministic Programs

Put simply, a program is control deterministic if and only if the control flow of the program is unaffected by the stochastic or nondeterministic choices made during the program execution. In effect, the program does not have any if-then-else branches, and all loops in the program terminate after a predetermined number of iterations. Furthermore, the "primitive" assignment statements of the program involve a continuous function as their RHS.

Formally, a control deterministic program over real-valued program vari-

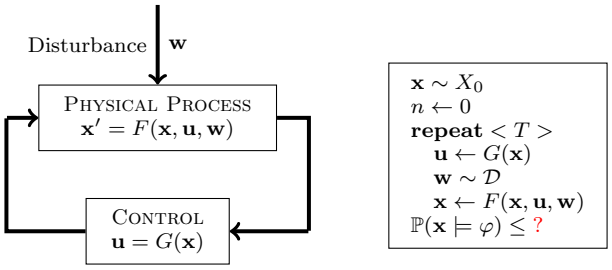


Figure 3 Discrete-time control of a physical process under uncertainties caused by external disturbances and a control deterministic probabilistic program that simulates it.

ables $\mathbf{x}$ is constructed using the grammar shown below:

$$
\begin{array}{rcl}
\text{program} & \rightarrow & \text{statement}^* \\
\text{statement} & \rightarrow & \text{assignment} \\
 & | & \textbf{repeat} < n > (\text{statement}^*) \\
\text{assignment} & \rightarrow & x_i \leftarrow f(x_{i_1}, \ldots, x_{i_k}) \\
 & | & x_j \sim \mathcal{D} \\
x_1, \ldots, x_n & \in & \text{Identifiers} \\
n & \in & \mathbb{N} \\
f & \in & \text{Continuous} \\
\mathcal{D} & \in & \text{Distributions}
\end{array}
$$

The program consists of a set Identifiers of real-valued state variables $x_1, \ldots, x_n$ that are manipulated using a sequence of assignment statements and deterministic loops that repeat a set of statements a fixed number n of times. Further, each assignment involves a continuous function f applied to a subset of variables. The statement $x \sim \mathcal{D}$ denotes drawing a sample from a distribution $\mathcal{D}$ and assigning the value to variable x. The semantics of such a program can be defined in the usual manner (see Kozen (1981)), and are omitted for this discussion.

Despite the limitations on expressivity due to the absence of control branches, control deterministic computations form an important class of probabilistic programs. They arise naturally in the domain of cyber-physical systems, wherein it is important to reason about uncertainty in the physical state of the system due to external disturbances. For instance, all the motivating examples from Section 1.1 are all control deterministic.

Figure 3 shows a schematic diagram of a physical process whose internal state $\mathbf{x}$ is updated at each time step using the law $\mathbf{x}' = F(\mathbf{x}, \mathbf{u}, \mathbf{w})$ wherein $\mathbf{u}$ is the control applied externally by a controller and $\mathbf{w} \sim \mathcal{D}$ is a stochastic disturbance. We assume that F is a continuous, but possibly nonlinear

function. Similarly, the feedback law G is a continuous and possibly nonlinear function $\mathbf{u} = G(\mathbf{x})$. Given the uncertainty in the initial state $\mathbf{x} \sim X_0$, our goal is to evaluate bounds on the probability that $\mathbf{x}(T) \models \varphi$ for some assertion φ specifying the unsafe set of states.

4.2 Symbolic Execution Using Affine Forms

In this section, we briefly describe an approach that symbolically executes a control deterministic program based on affine forms defined in previous work by Bouissou et al. (2012) and subsequently by Bouissou et al. (2016). Affine forms abstract how the variables in a computation depend as an affine function of the distributions that affect the program execution. However, many programs of interest are not affine. To handle these, affine forms are abstracted in two ways: (a) affine forms involve *abstract noise symbols* that represent a set of possible distributions; (b) the symbols in the affine form can be correlated.

Let us define a set of noise symbols $Y = \{y_1, y_2, \ldots\}$, wherein each symbol y_i has an associated set of support in the form of an interval $[\ell_i, u_i]$, intervals for expectation $\mathbb{E}(y_i) \in [a_i, b_i]$, and possibly, a list of intervals for its higher moments $\mathbb{E}(y_i^2), \mathbb{E}(y_i^3), \cdots, \mathbb{E}(y_i^k)$.

Definition 1.16 (Environment) An environment $\mathcal{E} := \langle Y, \mathsf{support}, \mathbb{E}(\cdot), G\rangle$ is given by a finite set of noise symbols $Y = \{y_1, \ldots, y_n\}$, a map $\mathsf{support}$ from each symbol y_j to an interval I_j indicating its set of support, a map that associates some select monomial terms $m := y_1^{k_1} \cdots y_n^{k_n}$ to intervals that bound their expectations $\mathbb{E}(m)$, and finally, a directed graph G whose vertices are the symbols in Y and edges (y_i, y_j) denote that the variable y_j is derived as a function of y_i (and possibly other variables in Y).

An environment $\mathcal{E}$ represents a set of distributions $\mathcal{D}$ over the noise symbols in Y such that the sets of support and expectations all lie in the intervals defined by the environment. The graph G defines the functional dependence or independence within pairs of random variables using the following definition.

Definition 1.17 (Probabilistic Dependence) Noise symbols y_i and y_j are *probabilistically dependent* random variables if there exists y_k such that there are paths from y_k to y_i and y_j to y_k in the graph G. Otherwise, y_i, y_j represent mutually independent random variables.

An environment $\mathcal{E}$ with noise symbols $\mathbf{y} := (y_1, \ldots, y_n)$ corresponds to a set of possible random vectors $Y := (Y_1, \ldots, Y_n)$ that conform to the

following constraints: (a) $(Y_1, \ldots, Y_n)$ must range over the set of support $\mathsf{support}(y_1) \times \cdots \times \mathsf{support}(y_n)$; (b) the moment vectors lie in the appropriate ranges defined by the environment; and, (c) if noise symbols y_i, y_j are probabilistically independent according to the dependence graph G, the corresponding random variables Y_i, Y_j are mutually independent. Otherwise, they are "arbitrarily" correlated while still respecting the range and moment constraints above.

Given an environment $\mathcal{E}$, affine forms are affine expressions over its noise symbols.

Definition 1.18 (Affine Forms) An affine form over an environment $\mathcal{E}$ is an expression of the form

$$a_0 + a_1 y_1 + \cdots + a_n y_n$$

where $a_0, a_1, \ldots, a_n$ are interval coefficients, and $y_1, \ldots, y_n$ are the corresponding noise symbols.

We assume that $\mathsf{support}(y_j)$ is bounded for all $y_j \in Y$. We, however, handle variables with unbounded set of support through the truncation procedure described in section 3.2. Another important aspect is that of missing moment information. We may use interval arithmetic to estimate missing information given the information on the set of support and available moments.

Lemma 1.19 *Let X be a (univariate) random variable whose set of support is the interval $I \subseteq \mathbb{R}$. It follows that $\mathbb{E}(X) \in I$.*

Let X_1, X_2 be two random variables. The following inequality holds:

$$-\sqrt{\mathbb{E}(X_1^2)\mathbb{E}(X_2^2)} \leq \mathbb{E}(X_1 X_2) \leq \sqrt{\mathbb{E}(X_1^2)\mathbb{E}(X_2^2)}\,.$$

The inequality above follows from the Cauchy-Schwarz inequality. Further details on how missing moment information is inferred are explained in Bouissou et al. (2016).

Example 1.20 First we will provide an illustrative example of an environment $\mathcal{E}$. Let $Y = \{y_1, y_2, y_3\}$ be a set of noise symbols such that $\mathsf{support}(y_1) = [-1, 1]$, $\mathsf{support}(y_2) = [0, 2]$ and $\mathsf{support}(y_3) = [-2, 3]$. The corresponding expectations are

$$\mathbb{E}(y_1) = \ [-0.1, 0.1],\ \mathbb{E}(y_2) = [1.1, 1.3],\ \mathbb{E}(y_3) = [-0.5, -0.3].$$

Furthermore, assume we are provided the higher order moment information

$$\mathbb{E}(y_1^2) = [0.2, 0.5],\ \mathbb{E}(y_1 y_2) = [-0.4, 0.6],\ \mathbb{E}(y_3^2) = [0.4, 0.6].$$

The dependency graph has the edges (y_1, y_3) indicating that y_3 is functionally dependent on y_1, which in turn are both pairwise independent of y_2.

An example affine form in this environment $\mathcal{E}$ is

$$\eta_1 := [0.5, 1.5] + [2.0, 2.01]y_1 - [2.8, 3.2]y_3 \,.$$

Semantically, an affine form $f(\mathbf{y}) := a_0 + \sum_{i=1}^{n} a_i y_i$ represents a set of linear expressions $[\![f(\mathbf{y})]\!]$ over $\mathbf{y}$:

$$[\![f(\mathbf{y})]\!] := \left\{ r_0 + \sum_{i=1}^{n} r_i Y_i \mid r_i \in a_i, (Y_1, \ldots, Y_n) \in [\![\mathcal{E}]\!] \right\} .$$

Given affine forms, we can define a calculus that describes how basic operations such as sums, differences, products and application of continuous (and k-times differentiable) functions are carried out over these affine forms.

Sums, Differences and Products: Let f_1, f_2 be affine forms in an environment $\mathcal{E}$ given by $f_1 := \mathbf{a}^t\mathbf{y} + a_0$ and $f_2 := \mathbf{b}^t\mathbf{y} + b_0$. We define the sum $f_1 \oplus f_2$ to be the affine form $(\mathbf{a} + \mathbf{b})^t\mathbf{y} + (a_0 + b_0)$. Likewise, let λ be a real number. The affine form λf_1 is given by $(\lambda\mathbf{a})^t\mathbf{y} + \lambda a_0$.

We now define the product of two forms $f_1 \otimes f_2$.

$$f_1 \otimes f_2 \;=\; a_0b_0 + a_0f_2 + b_0f_1 + \mathsf{approx}(\sum_{i=1}^{n}\sum_{j=1}^{n} a_ia_jy_iy_j) \,.$$

Note that a_0b_0, a_0f_2, b_0f_1 and a_ia_j denote the result of multiplying two intervals. The product of two intervals $[l_i, u_i][l_j, u_j]$ is defined as the interval $[\min(l_il_j, u_il_j, l_iu_j, u_il_j), \max(l_il_j, u_il_j, l_iu_j, u_il_j)]$ (see Moore et al. (2009)).

The product of two affine forms $f_1 \otimes f_2$ separates the affine and linear parts of this summation from the nonlinear part that must be approximated to preserve the affine form. To this end, we define a function approx that replaces the nonlinear terms by a collection of fresh random variables. In particular, we add a fresh random variable y_{ij} to approximate the product term y_iy_j.

Dependencies: We add the dependency edges (y_{ij}, y_i) and (y_{ij}, y_j) to the graph G to denote the newly defined functional dependences.

Set of Support: The set of support for y_{ij} is the interval product of the set of supports for y_i, y_j, respectively. In particular if $i = j$, we compute the set of support for y_i^2. Interval I_{ij} will represent the set of support for y_{ij}.

Moments: The moments of y_{ij} are derived from those of y_i and y_j, as follows.

Case-1 $(i = j)$. If $i = j$, we have that the $\mathbb{E}(y_{ij}^p) = \mathbb{E}(y_i^{2p})$. Therefore, the even moments of y_i are taken to provide the moments for y_{ij}. However, since we assume that only the first k moments of y_i are available, we have that

the first $\frac{k}{2}$ moments of y_{ij} are available, in general. To fill in the remaining moments, we approximate using intervals as follows: $\mathbb{E}(y_{ij}^r) \in I_{ij}^r$. While this approximation is often crude, this is a tradeoff induced by our inability to store infinitely many moments for the noise symbols.

Case-2 ($i \neq j$). If $i \neq j$, we have that $\mathbb{E}(y_{ij}^p) = \mathbb{E}(y_i^p y_j^p)$. If y_i, y_j form an independent pair, this reduces back to $\mathbb{E}(y_i^p)\mathbb{E}(y_j^p)$. Thus, in this instance, we can fill in all k moments directly as entry-wise products of the moments of y_i and y_j. Otherwise, they are dependent, so we use the Cauchy-Schwarz inequality (see Lemma 1.19): $-\sqrt{\mathbb{E}(y_i^{2p})\mathbb{E}(y_j^{2p})} \leq \mathbb{E}(y_{ij}^p) \leq \sqrt{\mathbb{E}(y_i^{2p})\mathbb{E}(y_j^{2p})}$, and the interval approximation $\mathbb{E}(y_{ij}^p) \in I_{ij}^p$.

Continuous Functions: Let $g(\mathbf{y})$ be a continuous and $(m+1)$-times differentiable function of $\mathbf{y}$, wherein $\mathbf{y}$ belongs to a compact interval J. The Taylor expansion of g around a point $\mathbf{y}_0 \in \mathsf{interior}(J)$ allows us to approximate g as a polynomial.

$$g(\mathbf{y}) = g(\mathbf{y}_0) + Dg(\mathbf{y}_0)(\mathbf{y} - \mathbf{y}_0) + \sum_{2 \leq |\alpha|_1 \leq m} \frac{D^\alpha g(\mathbf{y}_0)(\mathbf{y} - \mathbf{y}_0)^\alpha}{\alpha!} + R_g^{m+1},$$

wherein Dg denotes the vector of partial derivatives $(\frac{\partial g}{\partial y_j})_{j=1,\ldots,n}$, $\alpha := (d_1, \ldots, d_n)$ ranges over all vector of indices where $d_i \in \mathbb{N}$ is a natural number, $|\alpha|_1 := \sum_{i=1}^n d_i$, $\alpha! = d_1!d_2! \cdots d_n!$, $D^\alpha g$ denotes the partial derivative $\frac{\partial^{d_1} g \cdots \partial^{d_n} g}{\partial y_1^{d_1} \cdots \partial y_n^{d_n}}$ and $(\mathbf{y} - \mathbf{y}_0)^\alpha := \prod_{j=1}^n (y_j - y_{0,j})^{d_j}$. Finally, R_g^{m+1} is an interval valued *Lagrange remainder*:

$$R_g^{m+1} \in \left\{ \sum_{|\alpha|_1 = m+1} \frac{D^\alpha g(\mathbf{z})}{\alpha!} (\mathbf{z} - \mathbf{y}_0)^{m+1} \mid \mathbf{z} \in J \right\}.$$

This computation is automated in our implementation through a combination of standard ideas from automatic differentiation and interval arithmetic (Cf. Moore et al. (2009)).

Since we have discussed sums and products of affine forms, the Taylor approximation may be evaluated entirely using affine forms.

The remainder is handled using a fresh noise symbol $y_g^{(m+1)}$. Its set of support is R_g^{m+1} and moments are estimated based on this interval. The newly added noise symbol is functionally dependent on all variables $\mathbf{y}$ that appear in $g(\mathbf{y})$. These dependencies are added to the graph G.

The Taylor expansion allows us to approximate continuous functions including rational and trigonometric functions of these random variables.

Example 1.21 We illustrate this by computing the sine of an affine form.

Let y_1 be a noise symbol over the interval $[-0.2, 0.2]$ with the moments

$$(\mathbb{E}(y_1) = 0, \mathbb{E}(y_1^2) \in [0.004, 0.006], \mathbb{E}(y_1^3) = 0, \mathbb{E}(y_1^4) \in [6\times10^{-5}, 8\times10^{-5}], \mathbb{E}(y_1^5) = 0)\,.$$

We consider the form $\sin(y_1)$. Using a Taylor series expansion around $y_1 = 0$, we obtain

$$\sin(y_1) = y_1 - \frac{1}{3!}y_1^3 + [-1.3 \times 10^{-5}, 1.4 \times 10^{-5}]\,.$$

We introduce a fresh variable y_2 to replace y_1^3 and a fresh variable y_3 for the remainder interval $I_3 := [-1.3 \times 10^{-5}, 1.4 \times 10^{-5}]$.
Dependencies: We add the edges (y_2, y_1) and (y_3, y_1) to G.
Sets of Support: $I_2 := [-0.008, 0.008]$ and $I_3 := [-1.3 \times 10^{-5}, 1.4 \times 10^{-5}]$.
Moments: $\mathbb{E}(y_2) = \mathbb{E}(y_1^3) = 0$. Further moments are computed using interval arithmetic. The moment vector $I(m_2)$ is $(0, [0, 64\times10^{-6}], [-512\times10^{-9}, 512\times 10^{-9}], \ldots)$. For y_3, the moment vector

$$I(m_3) := (I_3, \text{square}(I_3), \text{cube}(I_3), \ldots)\,.$$

The resulting affine form for $\sin(y_1)$ is $[1,1]y_1 - [0.16, 0.17]y_2 + [1,1]y_3$.

4.3 Approximating Computations using Affine Forms

Having developed a calculus of affine forms, we may directly apply it to propagate uncertainties across control deterministic computations. Let $X = \{x_1, \ldots, x_p\}$ be a set of *program variables* collectively written as $\mathbf{x}$ with an initial value $\mathbf{x}_0$. Our semantics consist of a tuple $(\mathcal{E}, \eta)$ wherein $\mathcal{E}$ is an environment and $\eta := X \to \mathsf{AffineForms}(\mathcal{E})$ maps each variable $x_i \in X$ to an affine form over $\mathcal{E}$. The initial environment $\mathcal{E}_0$ has no noise symbols and an empty dependence graph. The initial mapping η_0 associates each x_i with the constant $x_{i,0}$. The basic operations are of two types: (a) assignment to a fresh random variable, and (b) assignment to a function over existing variables.

Random Number Generation: This operation is of the form $x_i := \mathsf{rand}(I, \mathbf{m})$, wherein I denotes the set of support interval for the new random variable, and $\mathbf{m}$ denotes a vector of moments for the generated random variable. The operational rule is $(\mathcal{E}, \eta) \xrightarrow{x_i := \mathsf{rand}(I,\mathbf{m})} (\mathcal{E}', \eta')$, wherein the environment $\mathcal{E}'$ extends $\mathcal{E}$ by a fresh random variable y whose set of support is given by I and moments by $\mathbf{m}$. The dependence graph is extended by adding a new node corresponding to y but without any new edges since freshly generated random numbers are assumed independent. However, if the

newly generated random variable is dependent on some previous symbols, such a dependency is also easily captured in our framework.

Assignment: The assignment operation is of the form $x_i \leftarrow g(\mathbf{x})$, assigning x_i to a continuous and $(j+1)$-times differentiable function $g(\mathbf{x})$. The operational rule has the form $(\mathcal{E}, \eta) \xrightarrow{x_i \leftarrow g(\mathbf{x})} (\mathcal{E}', \eta')$. First, we compute an affine form f_g that approximates the function $g(\eta(x_1), \ldots, \eta(x_n))$. Let Y_g denote a set of fresh symbols generated by this approximation with new dependence edges E_g. The environment $\mathcal{E}'$ extends $\mathcal{E}$ with the addition of the new symbols Y_g and and new dependence edges E_g. The new map is $\eta' := \eta[x_i \mapsto f_g]$.

Let $\mathcal{C}$ be a computation defined by a sequence of random number generation and assignment operations. Starting from the initial environment $(\mathcal{E}_0, \eta_0)$ and applying the rules above, we obtain a final environment $(\mathcal{E}, \eta)$. However, our main goal is to answer *queries* such as $\mathbb{P}(x_j \in I_j)$ that seek the probability that a particular variable x_j belongs to an interval I_j. This directly translates to a query involving the affine form $\eta(x_j)$ which may involve a prohibitively large number of noise symbols that may be correlated according to the dependence graph G.

Example 1.22 (2D robotic end effector) Consider a simplified version of the 2D robotic end effector model presented in Example 1.1, yielding an affine form with 6900 noise symbols for the variable x that we care about. The computation required 15 seconds of computational time on a laptop with Intel 3.1 core i7 processor and 16GB RAM.

$$x = \left\{ \begin{array}{c} [8.06365, 8.06441] + [1,1] * y_0 + [0.984807, 0.984808] * y_2 + \\ [0.0303060, 0.0303069] * y_3 + [-1,-1] * y_4 + \\ [0.0303060, 0.0303069] * y_5 + [-1,-1] * y_6 + \\ [0.499997, 0.500026] * y_9 + \\ [0.90686, 0.906894] * y_{10} + \\ \cdots \\ [0.119382, 0.119386] * y_{6885} + [-1,-1] * y_{6886} + [0.984807, 0.984808] * y_{6889} \\ + [0.0303060, 0.0303069] * y_{6890} + [-1,-1] * y_{6891} + [0.0303060, 0.0303069] * y_{6892} + \\ [-1,-1] * y_{6893} + [1,1] * y_{6896} + [-1,-1] * y_{6898} + [-1,-1] * y_{6899} \end{array} \right.$$

Based on the affine form, we can bound the support for $x \in [213.19, 326.12]$ and its expectation as $\mathbb{E}(x) \in [268.61, 270.7]$, and the second central moment (variance) in the range $[0.12, 0.28]$.

4.4 Applying Concentration of Measure Inequalities

We will now apply the results from section 3 to analyzing the affine forms generated from control deterministic programs. First, we note that each

affine form is a sum of possibly dependent random variables with information about sets of support, first and possibly higher order moments available. Thus, many strategies for applying the results in the previous section are available. These are summarized in detail in Bouissou et al. (2016). In what follows, we will illustrate the application of these results directly to some of the motivating examples from Section 1 using a prototype implementation of the ideas mentioned thus far. The prototype implementation in the C++ language interprets a given program using a library of affine forms. Next, it mechanizes the process of answering queries by analyzing the dependency graph. The automatic analysis uses a series of approaches that include:

(i) The application of Chernoff-Hoeffding bounds by using a *compaction* procedure that combines multiple noise symbols into a single one, so that the affine forms are all summations over independent random variables. Similarly, Bernstein inequalities are used whenever second moments are consistently available.
(ii) The application of Janson (2004) chromatic number bound, using $1+\Delta$ as an approximation for the fractional chromatic number, wherein Δ is the maximum degree of any node in the dependence graph.

Example 1.23 (2D end effector) Resuming the analysis in Ex. 1.22, we can automate the application of various approaches discussed thus far, starting with the Chernoff-Hoeffding bounds.

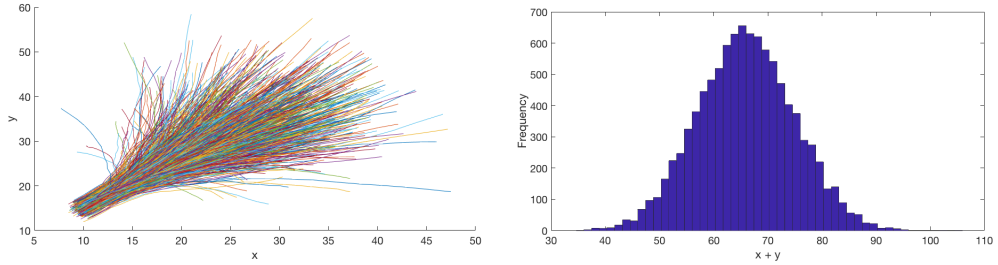
The original affine form has 6900 variables which are not all mutually independent. To obtain mutual independence, we analyze the strongly connected components of the undirected dependence graph yielding 3100 different components such that variables in distinct components are pairwise independent. Using this, we compact the affine form into one involving 3100 random variables and apply Chernoff-Hoeffding bounds. This is performed by computing the strongly connected components (SCC) of the dependency graph G, and taking the set of support and mean of the sum of random variables belonging to each SCC. Note that Chernoff-Hoeffding bounds can be applied since noise symbols belonging to different SCCs are mutually independent.

This yields

$$\mathbb{P}(x \geq 277) \leq \exp\left(\frac{-(268.6170484 - 277)^2}{7.486493141}\right) \leq 8.38 \times 10^{-5}\,.$$

Applying Bernstein's inequality yields:

$$\mathbb{P}(X \leq t) \leq \exp\left(\frac{-(268.6170484 - t)^2}{0.4868099186 + 0.3333 * (t - 286.6170484)}\right) \leq 5.18{\times}10^{-10}\,.$$



(a) **(b)**

Figure 4 **(a)** Simulated (x, y) positions of UAV over the time horizon $[0, 3.6]$ seconds and **(b)** Histogram of final position $x(3.6) + y(3.6)$.

The Chromatic number bound of Janson (2004) computes a weaker bound given by 0.106.

Now, we will consider the example of a fixed-wing UAV collision probability estimation from Ex. 1.3.

Example 1.24 Consider a prediction horizon of $t = 20 \times 0.18 = 3.6$ seconds. Our goal is to run the model twenty times, starting from a given initial state and query the probability that $x + y \geq 165$. We obtain an affine form for $x + y$ with 88 noise symbols. The mean value $\mathbb{E}(x + y) \in [65.85, 65.86]$ matches very well with the empirical estimate of 65.84 from $10,000$ simulations. Furthermore, the variance is estimated in the range $[78.95, 78.96]$ which also matches quite well with the empirical variance of 78.76 obtained from $10,000$ simulations. Some of the trajectories of the system and the scatter plot with 10000 end points are shown in Figure 4.

Using the Bernstein inequality, we obtain the estimate

$$\mathbb{P}(x + y \geq 165) \leq 9.6 \times 10^{-4}\,.$$

and more generally,

$$\mathbb{P}(x + y \geq 65.859 + t) \leq \exp\left(\frac{-t^2}{157.8869 + 12.57t}\right)\,.$$

5 Supermartingales and Concentration of Measure

In the final section, we look at concentration of measure inequalities using super-martingales. A previous chapter in the same volume by Chatterjee et al adapts the concept of a super-martingales to prove termination. We

will recall the definition and show that super-martingales are also useful for proving concentration. First let us recall conditional expectations. Let X, Y be two random variables. The conditional expectation $\mathbb{E}(X|Y)$ is defined as a function $f(y)$ defined over the support of the distribution Y such that

$$f(y) = \int_X x d\mathbb{P}(x|y)$$

In other words, for each value of y, the expectation integrates x over the conditional distribution of x given y.

Definition 1.25 (Martingales, Super- and Sub-Martingales) A sequence of random variables $X_0, X_1, \ldots, X_n$ is a *martingale* iff for each $i \geq 0$,

$$\mathbb{E}(X_{i+1} \mid X_i, \ldots, X_0) = X_i \,.$$

A supermartingale satisfies the condition

$$\mathbb{E}(X_{i+1} \mid X_i, \ldots, X_0) \leq X_i \,.$$

A *submartingale* satisfies the inequality:

$$\mathbb{E}(X_{i+1} \mid X_i, \ldots, X_0) \geq X_i \,.$$

A martingale is, therefore, both a supermartingale and a submartingale. Typically, the stochastic processes that are studied arise from Markovian models such as probabilistic programs wherein the next state distribution depends on just the current state. Thus, the conditional expectation $\mathbb{E}(X_{i+1} \mid X_i, \ldots, X_0)$ is written as $\mathbb{E}(X_{i+1} \mid X_i)$.

Example 1.26 Consider a random walk involving $x(t) \in \mathbb{Z}$ that is updated as

$$x(t+1) = \begin{cases} x(t)+1 & \text{with probability } \frac{1}{2} \\ x(t)-1 & \text{with probability } \frac{1}{2} \end{cases}$$

It is easy to see that $x(t)$ is a martingale since

$$\mathbb{E}(x(t+1) \mid x(t)) = \frac{1}{2}(x(t)+1) + \frac{1}{2}(x(t)-1) = x(t) \,.$$

It is easy to see that a martingale is always a supermartingale, but not necessarily vice-versa. Another important observation is that often a stochastic process is not a (super) martingale itself. However, another process built, for instance, by computing a function of the original process forms a (super) martingale.

Example 1.27 Consider a different scenario wherein $x(t) \in \mathbb{R}$.

$$x(t+1) = \begin{cases} 0.8x(t) & \text{with probability } \frac{1}{2} \\ 1.1x(t) & \text{with probability } \frac{1}{2} \end{cases}$$

$x(t)$ is neither a martingale or a super martingale. However, note that $y(t) = x(t)^2$ is a super martingale.

$$\mathbb{E}(y(t+1) \mid x(t)) = \frac{1}{2}0.8^2 y(t) + \frac{1}{2}1.1^2 y(t) = 0.925y(t) \leq y(t)\,.$$

Some of the constructions that have been previously encountered such as the McDiarmid's Inequality (Theorem 1.11) involve a martingale under the hood.

Example 1.28 (Doob Martingale) Let $f(x_1, \ldots, x_n)$ be a function with n inputs which are drawn from independent random variables $X_1, \ldots, X_n$.

Consider the stochastic process

$$Y_i = \mathbb{E}_{X_{i+1},\ldots,X_n}(f(X_1, \ldots, X_i, X_{i+1}, \ldots, X_n))\,,$$

for $i = 0, \ldots, n$. Note that each Y_i is a function of $X_1, \ldots, X_i$ while taking expectations over the remaining arguments. As a result Y_0 is the expected value of f under all its inputs, Y_i for $i > 0$ fixes random samples for the arguments indexed from 1 to i, and Y_n is the function f computed over some random sample of all the arguments.

Note that for every $i < n$, it is easy to show that

$$\mathbb{E}(Y_{i+1} \mid X_i, \ldots, X_1) = Y_i\,.$$

This construction can be achieved for any function f and is called Doob martingale. However, also note the independence requirements for the random variables $X_1, \ldots, X_n$.

Super-martingales from programs: As previously noted in chapter on termination, we seek expressions involving variables of the programs that form super-martingales.

Consider the program shown in Figure 5 (taken from our previous work Chakarov and Sankaranarayanan (2013)), wherein the position of an underwater vehicle (x, y) is updated at each step through a command that can be randomly chosen direction or just staying in one position. Based on this command, the actual position changes through a noisy execution of the command. However, at the same time, the estimation of the current position is updated. The question is how far the estimate deviates from the true position after

```
x, y, estX, estY = 0, 0, 0, 0
dx, dy, dxc, dyc = 0, 0, 0, 0
N = 500
for i in range(N):
    cmd = choice {N:0.1, S:0.1, E:0.1, W:0.1, NE:0.1, SE:0.1, NW: 0.1,
    SW: 0.1, Stay: 0.2}
    if (cmd == 'N'):
        dxc, dyc = 0, Uniform(1,2)
    elif (cmd == 'S'):
        dxc, dyc = 0, Uniform(-2, -1)
    elif (cmd == 'E'):
        dxc, dyc = Uniform(1,2), 0
    ...
    else // cmd == 'Stay'
        dxc, dyc = 0,0
    dx = dxc + Uniform(-0.05, 0.05)
    dy = dyc + Uniform(-0.05, 0.05)
    x = x + dx
    y = y + dy
    estX = estX + dxc
    estY = estY + dyc
assert( |x - estX| >= 3)
```

Figure 5 Program simulating a sequence of moves by a submarine, where (x, y) model the true position, dxc, dyc model the commanded change in position at any step, and $(estX, estY)$ model the estimates through dead-reckoning.

$N = 500$ steps? Note that for this program it is straightforward to establish that $x - estX$ and $y - estY$ are supermartingales.

Azuma-Hoeffding's Inequality: Let $\{X_n\}_{n=0}^{N}$ be a super martingale that satisfies a bounded difference condition that $|X_{i+1} - X_i| \leq c_i$ for each $i \in \{0, \ldots, N-1\}$. It follows that for any $j \in \{0, \ldots, N\}$,

$$\mathbb{P}(X_j - X_0 \geq t) \leq \exp\left(\frac{-t^2}{2\sum_{i=0}^{j-1} c_i^2}\right).$$

Furthermore, if X_n is a sub-martingale, we can conclude that

$$\mathbb{P}(X_j - X_0 \leq -t) \leq \exp\left(\frac{-t^2}{2\sum_{i=0}^{j-1} c_i^2}\right).$$

Thus, for a martingale which is a supermartingale as well as a submartingale, both inequalities hold.

The Azuma-Hoeffding bound is a concentration of measure inequality much along the lines of previously encountered inequalities in this chapter. For a martingale, it bounds the probability of a large deviation on either side of its starting value. For a super-martingale, the inequality bounds the probability of a large deviation above the starting value. The martingale condition generalizes from the need for independent random variables that

was seen for the case of Chernoff-Hoeffding inequalities. Just as the latter inequalities are applied to random variables with bounded sets of support, we note the condition of bounded change on the (super) martingale.

Example 1.29 Returning to the dead-reckoning example, we can use the martingale $x - estX$ to estimate the failure probability of the assertion at the end of the program. We note that every loop iteration, the absolute change in this expression is bounded by 0.05. Therefore, we obtain

$$\mathbb{P}(|x - estX| \geq 3) \leq 1.5 \times 10^{-3}\,.$$

Identical bounds are obtained for the deviation of y from $estY$, as well.

Super martingale approaches form very powerful approaches for quantitative reasoning. However, two important roadblocks exist to their wider application:

(i) Automatically discovering super martingale expressions remains a hard problem, especially for nonlinear expressions. However, a variety of approaches summarized in the termination chapter can be used in this regard. At the same time, the applications have been limited thus far.
(ii) The resulting bounds remain conservative since independent random variables are often treated as possibly dependent in the analysis for supermartingales. For instance, treating $x - estX$ as the sum of independent random variables for the previously considered dead-reckoning example yields much more precise bounds.

However, super martingales remain a promising approach for quantitative reasoning for more complex models that involve programs with branch conditions that cannot be treated with the approaches for control deterministic computations reviewed in the previous sections.

6 Conclusion

Thus far, we have examined situations where concentration of measure inequalities can be applied to analyze probabilistic programs. As the reader may have noticed, the key issue lies in mechanizing the process of inference, since even small programs can lead to cumbersome calculations that are hard to carry out by hand. However, there are numerous challenges that must be tackled before the full power of these approaches can be realized. First, most approaches are restricted to reasoning about programs that manipulate real values, whereas programs can exhibit a rich variety of structures ranging

from Booleans, strings, lists, trees and graphs. Extending the concentration of measure approach to reason about a richer set of programs is an important area of future work.

Also, one notes that concentration of measure inequalities are often derived to uniformly exploit available moment information such as first moments, second moments and so on. It is easy to envision a process of customization that can derive inequalities based on the available moment information on a "per-problem" basis. This approach of "designing" new inequalities on demand is yet another promising area of future investigation.

Finally, the broader area of analyzing probabilistic programs has been seemingly disconnected from the problem of Bayesian inference, which is an important concern for probabilistic programming. We note that the problem of model inference and analysis are important steps in the overall "analytics" pipeline. It is therefore natural, as a first step to study these problems separately. The problem of integrating Bayesian inference and subsequent analysis of the "posterior" model/program remains an important unsolved problem. Current approaches that combine Monte-Carlo techniques with their approximate convergence guarantees are not directly compatible with the use of concentration of measure or other symbolic approaches presented here. In this regard, the study of imprecise models of probability distributions, representing sets of distributions, along with concentration of measure inequalities on functions of samples drawn from such models is another promising area of future investigation.

Acknowledgments: The author gratefully acknowledges collaboration with Aleksandar Chakarov, Olivier Bouissou, Eric Goubault and Sylvie Putot that formed the basis for the ideas presented in this chapter. We acknowledge support from the US National Science Foundation (NSF) under awards 1320069, 1527075 and 1815983. All opinions expressed are those of the authors and not necessarily of the NSF

Bibliography

Agha, Gul, and Palmskog, Karl. 2018. A Survey of Statistical Model Checking. *ACM Trans. Model. Comput. Simul.*, **28**(1), 6:1–6:39.

Bernstein, Sergei N. 1924. On a modification of Chebyshevs inequality and of the error formula of Laplace. *Ann. Sci. Inst. Sav. Ukraine, Sect. Math*, **1**.

Boucheron, Stephane, Lugosi, Gabor, and Massart, Pascal. 2016. *Concentration of Measure: An Asymptotic Theory of Independence*. Oxford University Press.

Bouissou, Olivier, Goubault, Eric, Goubault-Larrecq, Jean, and Putot, Sylvie. 2012. A generalization of P-Boxes to affine arithmetic. *Computing*.

Bouissou, Olivier, Goubault, Eric, Putot, Sylvie, Chakarov, Aleksandar, and Sankaranarayanan, Sriram. 2016. Uncertainty Propagation Using Probabilistic Affine Forms and Concentration of Measure Inequalities. Pages 225–243 of: Chechik, Marsha, and Raskin, Jean-François (eds), *Tools and Algorithms for the Construction and Analysis of Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Bournez, Olivier, and Garnier, Florent. 2005. Proving Positive Almost-Sure Termination. Pages 323–337 of: *RTA*. LNCS, vol. 3467. Springer.

Brockwell, P. J., and Davis, R. A. 2009. *Time Series: Theory and Methods*. Springer, New York.

Bucklew, James Antonio. 2004. *Introduction to Rare-Event Simulations*. Springer.

Chakarov, Aleksandar, and Sankaranarayanan, Sriram. 2013. Probabilistic Program Analysis using Martingales. Pages 511–526 of: *Computer-Aided Verification (CAV)*. Lecture Notes in Computer Science, vol. 8044. Springer-Verlag.

Chakarov, Aleksandar, Voronin, Yuen-Lam (Vris), and Sankaranarayanan, Sriram. 2016. Deductive Proofs of Almost Sure Persistence and Recurrence Properties. Pages 260–279 of: *Tools and Algorithms for Construction and Analysis of Systems (TACAS)*. Lecture Notes in Computer Science, vol. 9636. Springer-Verlag.

Chatterjee, Krishnendu, Novotný, Petr, and Zikelic, Dorde. 2017. Stochastic invariants for probabilistic termination. Pages 145–160 of: *ACM Principles of Programming Languages (POPL)*. ACM Press.

Chatterjee, Krishnendu, Fu, Hongfei, Novotný, Petr, and Hasheminezhad, Rouzbeh. 2018. Algorithmic Analysis of Qualitative and Quantitative Termination Problems for Affine Probabilistic Programs. *ACM Trans. Program. Lang. Syst.*, **40**(2), 7:1–7:45.

Clarke, Edmund, Donze, Alexandre, and Legay, Axel. 2009. Statistical Model Checking of Analog Mixed-Signal Circuits With An application

to a third order $\Delta - \Sigma$ modulator. Pages 149–163 of: *Hardware and Software: Verification and Testing*. LNCS, vol. 5394/2009.

Cousot, Patrick, and Monerau, Michael. 2012. Probabilistic Abstract Interpretation. Pages 169–193 of: *ESOP*. LNCS, vol. 7211. Springer.

De Loera, J., Dutra, B., Koeppe, M., Moreinis, S., Pinto, G., and Wu, J. 2011. Software for Exact Integration of Polynomials over Polyhedra. *ArXiv e-prints*, July.

Dempster, A.P. 1967. Upper and Lower Probabilities Induced by a Multivalued Mapping. *The Annals of Mathematical Statistics*, **38**(2), 325–339.

Dimitrova, Rayna, Fioriti, Luis María Ferrer, Hermanns, Holger, and Majumdar, Rupak. 2016. Probabilistic CTL*: The Deductive Way. Pages 280–296 of: *Proc. TACAS*.

Dubhashi, Devdatt, and Ranjan, Desh. 1998. Balls and Bins: A Study in Negative Dependence. *Random Structures and Algorithms*, **13**(2), 99–124.

Dubhashi, Devdutt, and Panconesi, Alessandro. 2009. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press.

Esparza, Javier, Gaiser, Andreas, and Kiefer, Stefan. 2012. Proving Termination of Probabilistic Programs Using Patterns. Pages 123–138 of: *CAV*. LNCS, vol. 7358. Springer.

Ferson, Scott, Kreinovich, Vladik, Ginzburg, Lev, Myers, David S., and Sentz, Kari. 2003 (January). *Constructing probability boxes and Dempster-Shafer structures*. Tech. rept. SAND2002-4015. Sandia Laboratories.

Fioriti, Luis María Ferrer, and Hermanns, Holger. 2015. Probabilistic Termination: Soundness, Completeness, and Compositionality. Pages 489–501 of: *Proc. Principles of Programming Languages, POPL*. ACM Press.

Geldenhuys, Jaco, Dwyer, Matthew B., and Visser, Willem. 2012. Probabilistic symbolic execution. Pages 166–176 of: *ISSTA*. ACM.

Hoeffding, Wassily. 1963. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, **58**(301), 13–30.

Janson, Svante. 2004. Large deviations for sums of partly dependent random variables. *Random Structures Algorithms*, **24**(3), 234–248.

Jégourel, Cyrille, Legay, Axel, and Sedwards, Sean. 2012. Cross-Entropy Optimisation of Importance Sampling Parameters for Statistical Model Checking. Pages 327–342 of: *CAV*. Lecture Notes in Computer Science, vol. 7358. Springer.

Jha, Sumit Kumar, Clarke, Edmund M., Langmead, Christopher James, Legay, Axel, Platzer, André, and Zuliani, Paolo. 2009. A Bayesian

Approach to Model Checking Biological Systems. Pages 218–234 of: *CMSB*. Lecture Notes in Computer Science, vol. 5688. Springer.

Kaminski, Benjamin Lucien, Katoen, Joost-Pieter, Matheja, Christoph, and Olmedo, Federico. 2016. Weakest Precondition Reasoning for Expected Run-Times of Probabilistic Programs. Pages 364–389 of: *European Symposium on Programming (ESOP)*.

Kass, Robert E., and Raftery, Adrian E. 1995. Bayes Factors. *J. Amer. Stat. Assoc.*, **90**(430), 774–795.

Katoen, Joost-Pieter, McIver, Annabelle, Meinicke, Larissa, and Morgan, Carroll. 2010. Linear-Invariant Generation for Probabilistic Programs. Page 390406 of: *Static Analysis Symposium (SAS)*. LNCS, vol. 6337. Springer.

Kozen, Dexter. 1981. Semantics of Probabilistic Programs. *J. Computer and System Sciences*, **22**, 328–350.

Kwiatkowska, M., Norman, G., and Parker, D. 2011. PRISM 4.0: Verification of Probabilistic Real-time Systems. Pages 585–591 of: *CAV*. LNCS, vol. 6806. Springer.

Mardziel, Piotr, Magill, Stephen, Hicks, Michael, and Srivatsa, Mudhakar. 2011 (JUN). Dynamic Enforcement of Knowledge-based Security Policies. Pages 114–128 of: *Computer Security Foundations Symposium (CSF)*.

McClain, Deborah A., and Hug, Carl C. 1980. Intravenous Fentanyl Kinetics. *Clinical Pharmacology & Therapeutics*, **28**(1), 106–114.

McDiarmid, Colin. 1989. On the method of bounded differences. *Surveys in combinatorics*, **141**(1), 148–188.

McElreath, Richard. 2015. *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. CRC Press.

McIver, Annabelle, and Morgan, Carroll. 2004. *Abstraction, Refinement and Proof for Probabilistic Systems*. Monographs in Computer Science. Springer.

McIver, Annabelle, Morgan, Carroll, Kaminski, Benjamin Lucien, and Katoen, Joost-Pieter. 2018. A new proof rule for almost-sure termination. *PACMPL*, **2**(POPL), 33:1–33:28.

Monniaux, David. 2000. Abstract Interpretation of Probabilistic Semantics. Pages 322–339 of: *Static Analysis Symposium (SAS)*. Lecture Notes in Computer Science, no. 1824. Springer Verlag.

Monniaux, David. 2005. Abstract interpretation of programs as Markov decision processes. *Sci. Comput. Program.*, **58**(1-2), 179–205.

Moore, R. E., Kearfott, R. B., and Cloud, M. J. 2009. *Introduction to Interval Analysis*. SIAM.

Narayanan, Praveen, Carette, Jacques, Romano, Wren, Shan, Chung-chieh, and Zinkov, Robert. 2016. Probabilistic inference by program transformation in Hakaru (system description). Pages 62–79 of: *International*

Symposium on Functional and Logic Programming - 13th International Symposium, FLOPS 2016, Kochi, Japan, March 4-6, 2016, Proceedings. Springer.

Parker, David, Norman, Gethin, and Kwiatkowska, Marta. 2006. Game-based Abstraction for Markov Decision Processes. Pages 157–166 of: *Quantitative Evaluation of Systems - (QEST'06)*. IEEE Press.

Rubinstein, Reuven Y., and Kroese, Dirk P. 2008. *Simulation and the Monte Carlo Method*. Wiley Series in Probability and Mathematical Statistics.

Sankaranarayanan, Sriram, and Fainekos, Georgios E. 2012. Falsification of temporal properties of hybrid systems using the cross-entropy method. Pages 125–134 of: *HSCC*. ACM.

Sankaranarayanan, Sriram, Chakarov, Aleksandar, and Gulwani, Sumit. 2013. Static analysis for probabilistic programs: inferring whole program properties from finitely many paths. Pages 447–458 of: *PLDI*. ACM.

Shafer, Glenn. 1976. *A Mathematical Theory of Evidence*. Princeton University Press.

Shafer, Steven L., Siegel, Lawrence C., Cooke, James E., and Scott, James C. 1988. Testing Computer-controlled Infusion Pumps by Simulation. *Anesthesiology*, **68**, 261–266.

Srinivasan, Rajan. 2002. *Importance Sampling: Applications in Communications and Detection*. Springer.

Wald, A. 1945. Sequential Tests of Statistical Hypotheses. *The Annals of Mathematical Statistics*, **16**(2), 117–186.

Williams, David. 1991. *Probability with Martingales*. Cambridge University Press.

Wingate, David, and Weber, Theophane. 2013. Automated Variational Inference in Probabilistic Programming. *CoRR*, **abs/1301.1299**.

Younes, Håkan L. S., and Simmons, Reid G. 2006. Statistical Probabilitistic Model Checking with a Focus on Time-Bounded Properties. *Information & Computation*, **204**(9), 1368–1409.

Yousefi, Mahdi, van Heusden, Klaske, M. Mitchell, Ian, Ansermino, Mark, and Dumont, Guy. 2017. A Formally-Verified Safety System for Closed-Loop Anesthesia. **50**(07), 4424–4429.