Abstraction of expectation functions using Gaussian distributions

David Monniaux

http://www.di.ens.fr/~monniaux
David.Monniaux@ens.fr
LIENS
45 rue d'Ulm
75230 Paris cedex 5, France

Abstract. We consider semantics of infinite-state programs, both probabilistic and nondeterministic, as expectation functions: for any set of states A, we associate to each program point a function mapping each state to its expectation of starting a trace reaching A. We then compute a safe upper approximation of these functions using abstract interpretation. This computation takes place in an abstract domain of extended Gaussian (normal) distributions.

Category: 1 (new results)

1 Introduction

Much progress has been made recently on the model-checking of probabilistic and nondeterministic systems, considered as Markov decision processes [5, 10, 7]. These methods, however, consider finite-state processes. For infinite state processes, one can either approximate them using finite-state processes, or use a symbolic approximation. In this paper, we take the latter approach.

1.1 Contribution

We propose a symbolic method for computing a safe upper-bound on the probability of a certain set of events occuring in the set of traces of the analyzed program. Our analysis method is set in the general framework of abstract interpretation [4] of probabilistic programs using expectation functions [14]. This analysis method approximates expectation functions from above using Gaussian extended (n-dimensional) distributions . The kind of results that can be established using this analysis is as follows:

$$\forall x \; E_A(\boldsymbol{x}) \leq \alpha \exp(-Q(\boldsymbol{x} - \boldsymbol{x}_0))$$

where $E_A(x)$ notes the expectation of reaching a state in A from an initial state x (a vector of real numbers representing the values of the various variables) and Q is a positive quadratic form. This result is achieved through an over-approximation

of the *value iteration* sequence associated with the Markov decision process [15] using *widening operators* to force convergence [4].

This result is *sound*, meaning that the bound that is obtained is necessarily true (although not optimal in general). The algorithms used are mostly standard numerical analysis, enabling the use of standard (bi)linear algebra packages such as Matlab or libraries such as Lapack [1].

1.2 Comparison with other works

The field of probabilistic model checking has developed considerably during the last few years. Some tools [5] are now available ; they use sophisticated algorithms on compact symbolic representations of vectors in $[0,1]^N$ where N is the number of states. On the other hand, these tools are unable to work directly on computer programs, which have infinite or at least very large state spaces (a system with forty 32-bit variables has about 10^{384} states); they need a preliminary step of (mostly manual) abstraction into a finite model.

In an earlier paper [14], we proposed a static analysis method for probabilistic and nondeterministic programs. That method used an abstract domain of step functions, that is, linear combinations of characteristic functions of basic elements. Those basic elements are taken in an abstract domain suitable for non-probabilistic analysis. That analysis method tends to perform well on the "big mass" of distributions, but gives overly coarse approximations of the "tails" of the expectation functions : they are uniformly bounded by a constant, whereas we would like to see them bounded by some function that is negligible far away.

We also proposed a method combining Monte-Carlo sampling and non-probabilistic abstract interpretation [12]. That method considers a slightly different semantics and achieves results that are valid only up to a confidence interval; it can become prohibitively expensive if the probability to be bounded with good relative accuracy is small. It is however possible to use analyses such as the one described in this paper to fine-tune the number of samples needed in different regions by the Monte-Carlo analysis.

1.3 Structure of the paper

In section 2, we shall explain briefly our notion of backward probabilistic abstract interpretation [12]. In section 3, we shall see the abstract domain of extended Gaussian distributions. In section 4, we shall see a few mathematical facts on second-degree polynomials and positive quadratic forms, as well as some effective algorithms.

2 Backwards probabilistic abstract interpretation

With respect to probabilistic program semantics, one has the choice between forward denotational semantics [8,9,11], backward denotational semantics [14] and small-step operational semantics (Markov decision processes) [13, ch. 7,8]. The

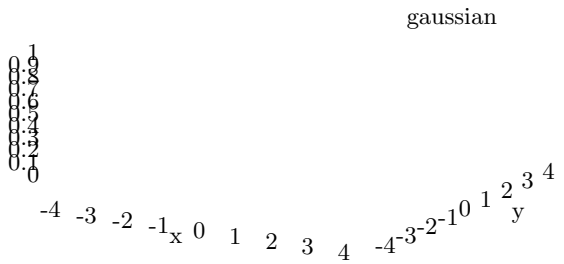

Fig. 1. An extended Gaussian distribution. Its matrix is $\left(\begin{smallmatrix} 0{,}6 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ in the orthogonal basis $\left(\begin{smallmatrix} \cos 0.4 & -\sin 0.4 \\ \sin 0.4 & \cos 0.4 \end{smallmatrix}\right)$.

analysis method described in this paper applies both to backward denotational semantics and backward small-step operational semantics. However, for the sake of brevity, we shall explain it using denotational semantics.

2.1 Concrete semantics

We shall begin by giving the *concrete semantics*, that is, the precise semantics of the programs to be analyzed.

In his seminal papers [8,9], Kozen introduced semantics of probabilistic programs as continuous linear operators on measure spaces. These semantics are *forward*, since they map the input probability distribution (measure) of the program to its output distribution. Using a linear duality relation, we obtained a backward probabilistic semantics operating on measurable functions [14]. This semantics is easily lifted to nondeterministic and probabilistic programs, which take choices, some of which according to a known probability distribution and the others in a certain known domain. We recall here this semantics in a compositional fashion.

$[\![H]\!]_p^*$ is the semantics of program construct H. If the environment (vector of variables, registers, heap...) before H is taken in the set X and after H taken in Y, then $[\![H]\!]_p^*$ is a function from $Y \to_{\text{measurable}} [0,1]$ to $X \to_{\text{measurable}} [0,1]$. This function is upper-continuous (the image of the limit of an ascending sequence is the limit of the images) and, when H does not include nondeterministic constructs, it is linear.

Sequence Straightforward composition:

$$[\![e_1\,;e_2]\!]_p^* = [\![e_1]\!]_p^* \circ [\![e_2]\!]_p^*. \tag{1}$$

Tests R_W is the linear operator mapping a function f to its pointwise product with the characteristic function of W. Then

$$[\![\texttt{if } c \texttt{ then } e_1 \texttt{ else } e_2]\!]_p^* = R_{\chi_{[\![c]\!]}} \circ [\![e_1]\!]_p^* + R_{\chi_{[\![c]\!]^C}} \circ [\![e_2]\!]_p^*. \tag{2}$$

Loops $\mathrm{lfp}\,\Psi$ is the least fixpoint of Ψ. Since Ψ is upper-continuous, $\mathrm{lfp} = \sqcup_n \Psi^n(0)$. Then

$$[\![\texttt{while } c \texttt{ do } e]\!]^*_p.f_0 = R_{\chi_{[\![c]\!]^C}}.\mathrm{lfp}\left(f \mapsto f_0 + R_{\chi_{[\![c]\!]}} \circ [\![e]\!]^*_p(f)\right) \tag{3}$$

This equation is the denotational version of the definition of the value of a Markov decision process using value iteration [15, §7.2.4].

Deterministic operations These are operations such as arithmetic, fetching data etc...

$$[\![\mathtt{H}]\!]^*_p.f = f \circ [\![\mathtt{H}]\!] \tag{4}$$

where $[\![\mathtt{H}]\!]$ is the denotational semantics of the deterministic operation H, mapping each input to the operation to the corresponding output.

Random generation The operation takes an environment (vector of variables) x and appends to it a random value `r` taken according to the random distribution μ_R.

$$([\![\texttt{r:=random}]\!]^*_p.f)(x) = \int f(x,r)\,\mathrm{d}\mu_R(r) \tag{5}$$

Nondeterministic generation The operation takes an environment (vector of variables) x and appends to it a variable `y` nondeterministically chosen in Y.

$$([\![\texttt{r:=nondeterministic}]\!]^*_p.f)(x) = \sup_{y \in Y} f(x,y) \tag{6}$$

2.2 Abstract interpretation

Following [14], we associate an abstract semantics $[\![H]\!]^{*\sharp}_p$ to any program construct H. This semantics is linked to the concrete one by the *abstraction relation* :

$$\forall f, f^\sharp\ f \leq f^\sharp \implies [\![H]\!]^*_p.f \leq [\![H]\!]^{*\sharp}_p.f^\sharp, \tag{7}$$

using the pointwise ordering.

We shall restrict the abstract computations to a certain family of functions taken in an *abstract domain*. We shall introduce a domain of extended Gaussian distributions in section 3.

The domain should implement some elementary abstract operations:

- abstract counterparts of the semantics of deterministic operations, nondeterministic generation, probabilistic generation
- an abstraction $+^\sharp$ of the $+$ operator
- an abstraction $R^\sharp_{[\![c]\!]}$ of $R_{[\![c]\!]}$ for any boolean condition c (such as a comparison between two variables)
- an abstraction of the $\sqcup$ (least upper bound operator):

$$f^\sharp \sqcup^\sharp g^\sharp \geq f^\sharp \tag{8}$$

$$f^\sharp \sqcup^\sharp g^\sharp \geq g^\sharp \tag{9}$$

- a *widening operator* ∇:

- for all $f^\sharp$ and $g^\sharp$, $f^\sharp \nabla g^\sharp$ is greater than $f^\sharp$ and $g^\sharp$;
- for any sequence $v_n^\sharp$ and any $u_0^\sharp$, the sequence defined by $u_{n+1}^\sharp = u_n^\sharp \nabla v_n^\sharp$ is ultimately stationary.

The widening operator ∇ is a kind of convergence accelerator for fixpoint iterations [4, §4.3]. Let us suppose we have a concrete function Ψ and its abstract counterpart $\Psi^\sharp$. We wish to obtain an abstraction (upper approximation) of $\operatorname{lfp} \Psi$ (the least fixpoint of Ψ). $\operatorname{lfp} \Psi$ is the limit of the sequence $(f_n)_{n\in\mathbb{N}}$ defined by $f_0 = 0$ and $f_{n+1} = \Psi f_n$. Let us now consider the sequence $f_0^\sharp = 0$ and $f_{n+1}^\sharp = f_n^\sharp \nabla \Psi^\sharp f_n^\sharp$. Obviously, $f_n \leq f_n^\sharp$ for any n. Furthermore, $f_n^\sharp$ is ultimately stationary. Its limit $L^\sharp$ is therefore an upper bound on $\operatorname{lfp} \Psi = \lim_{n\to n} f_n^\sharp$. Let us note it $\operatorname{lfp}^\sharp \Psi^\sharp$.

The abstract semantics is obtained by replacing each elementary operation in the definition of the concrete semantics (§2.1) by its abstract counterpart.

3 Extended Gaussian distributions

We shall now describe the domain of extended Gaussian distributions (Fig. 1).

3.1 Construction

We shall first describe the form of the functions making up the abstract domain.

Definition 1. *Let E be a finite dimensional real vector space. Let us consider a positive quadratic form Q and a linear form L over E such that* $\ker Q \subseteq \ker L$. *q_0 is a real number. The function*

$$\begin{aligned} E &\to \mathbb{R}_+ \\ \boldsymbol{v} &\mapsto \exp(-Q(\boldsymbol{v}) + L(\boldsymbol{v}) + q_0) \end{aligned}$$

is called an extended Gaussian distribution. *It shall be noted G_{Q,L,q_0}.*

Proposition 1. *Let $\phi : \boldsymbol{v} \mapsto \exp(-Q(\boldsymbol{v}) + L\boldsymbol{v} + q)$ be an extended Gaussian distribution over an euclidean space E. Then there exists an orthonormal basis $(\boldsymbol{v}_i)_{1\leq 1\leq n}$, a positive real number K, coefficients $(\lambda_i)_{1\leq 1\leq n}$ and coordinates $(p_i)_{1\leq 1\leq n}$ such that*

$$\phi\left(\sum x_i \boldsymbol{v}_i\right) = K \exp\left(-\sum_i \lambda_i (x_i - p_i)^2\right).$$

The point P, whose coordinates in the basis $(\boldsymbol{v}_i)_{1\leq 1\leq n}$ are $(p_i)_{1\leq 1\leq n}$, is called the center *of the distribution.*

3.2 Least upper bound and widening

Let (Q_1, L_1, q_1) and (Q_2, L_2, q_2) be two extended Gaussian distributions. We wish to get a common upper bound for them.

Let us note that, in general, there is no least upper bound in Gaussian distributions, even when the Gaussians are centered and unscaled: two ellipses with a common center do not necessarily have a least upper bound.

We define the extended Gaussian distribution $(Q_1, L_1, q_1) \sqcup (Q_2, L_2, q_2)$ as follows. Since Q_1 and Q_2 are positive, we diagonalize them in the same base $(\boldsymbol{v}_i)_{1\leq 1\leq n}$ (theorem 2). Then $Q_1(\sum_i x_i \boldsymbol{v}_i) = \sum \lambda_i x_i^2$ and $Q_2(\sum_i x_i \boldsymbol{v}_i) = \sum \mu_i x_i^2$. Let us write the linear forms L_1 and L_2 in this basis: $L_1(\sum_i x_i \boldsymbol{v}_i) = \sum \alpha_i x_i$ and $L_2(\sum_i x_i \boldsymbol{v}_i) = \sum \beta_i x_i$.

Let σ_i and τ_i be partitions of q_1 and q_2 respectively ($\sum_i \sigma_i = q_1$ and $\sum_i \tau_i = q_2$). We can take $\sigma_i = q_1/n$ and $\tau_i = q_2/n$.

Let $a_i X^2 + b_i X + c_i = (\lambda_i X^2 + \alpha_i X + \sigma_i) \sqcap (\mu_i X^2 + \beta_i X + \tau_i)$ (as defined in §4.1).

Let $Q(\sum_i x_i \boldsymbol{v}_i) = \sum_i a_i x_i^2$, $L(\sum_i x_i \boldsymbol{v}_i) = \sum_i b_i x_i$ and $q = \sum_i c_i$.

Let us check that $\ker Q \subseteq \ker L$. Since $\ker Q$ is the isotropic cone of Q and the $\boldsymbol{v}_i$ form a diagonal basis for Q, it follows that a subset of the $\boldsymbol{v}_i$ form a basis of $\ker Q$. For any index i such that $\boldsymbol{v}_i$ is in that basis, $a_i = 0$; since we exclude polynomials of degree one ($\ker Q \subseteq \ker L$), b_i must also be null.

by construction, so $\boldsymbol{v}_i \in \ker L$.

We define

$$(Q_1, L_1, q_1) \sqcup (Q_2, L_2, q_2) = (Q, L, q). \tag{10}$$

Let us remark that

$$\dim \ker Q \geq \max(\dim \ker Q_1, \dim \ker Q_2) \tag{11}$$

since for all i such that $\lambda_i = 0$ or $\mu_i = 0$, $a_i = 0$.

We define the widening operator similarly, except that this time we need to ensure convergence of the ascending sequences $u_{n+1} = u_n \nabla v_n$. We shall modify the least upper bound operator in two respects to obtain the widening operator:

1. Intuitively, when $a_i < \lambda_i$, this means that along that particular vector $\boldsymbol{v}_i$ the Gaussian gets flatter and flatter. The natural widening is to upper-approximate it by a flat line. In this case, we take $a'_i = b'_i = 0$ and

$$c'_i = \min_x(a_i x^2 + b_i x + c) = \frac{-b^2}{4a} + c. \tag{12}$$

2. If all $a_i = 0$ and c is still decreasing, we take the last resort of removing all constraints.

The convergence of the method is ensured by the fact that whenever step 1 is applied, $\dim \ker Q$ strictly increases. Since in non-widening steps, $\dim \ker Q$ increases or stays constant, it follows that at most $\dim E$ step 1 may be applied. After this, the a_i stay constant. The only way the sequence can still go on ascending is by an increase in c. Then step 2 ensures termination.

3.3 Random generators

Let us recall Equ. 5 the backwards operation associated with $\rho = \texttt{random}$ where ρ is a fresh real variable

$$g = [\![\rho = \texttt{random}]\!]_p^*.f = \boldsymbol{v} \mapsto \int_x f(\boldsymbol{v} + x\boldsymbol{e})\,\mathrm{d}\mu_R(x) \tag{13}$$

where $\boldsymbol{e}$ is an additional basis vector corresponding to the fresh variable ρ and μ_R is the distribution of the new variable. If μ_R is given by the Gaussian $\exp(-(\lambda x^2 + c_1))$, and f is given by (Q, L, c), then

$$\begin{aligned} g(\boldsymbol{v}) &= \int_{-\infty}^{+\infty} \exp\left(-(Q(\boldsymbol{v}+x\boldsymbol{e}) + L(\boldsymbol{v}+x\boldsymbol{e}) + c + \lambda x^2 + c_1)\right)\,\mathrm{d}x \\ &= \exp\left(-\left(\underbrace{Q(\boldsymbol{v}) - \frac{1}{4\alpha}Q^*(\boldsymbol{v},\boldsymbol{e})^2}_{Q'(\boldsymbol{v})} + \underbrace{L(\boldsymbol{v}) - \frac{1}{2\alpha}Q^*(\boldsymbol{v},\boldsymbol{e})L(\boldsymbol{e})}_{L'(\boldsymbol{v})} + \underbrace{c + c_1 - \frac{L(\boldsymbol{e})^2}{4\alpha} - \frac{1}{2}\log\frac{\pi}{a}}_{c'}\right)\right) \end{aligned} \tag{14}$$

Because of the definition of g as the integral of a bounded function versus a measure of finite total weight, g is bounded; thus Q is positive and $\ker L \subseteq \ker Q$.

3.4 Linear operations

We shall deal here with program statements such as $v_n := \sum_i \alpha_i v_i$ and more generally any linear transformation M where the vector of variables $\boldsymbol{V'}$ after the instruction is $M.\boldsymbol{V}$ where $\boldsymbol{V}$ is the vector of variable before the instruction. Following Equ. 4,

$$[\![\boldsymbol{V} := M.\boldsymbol{V}]\!]_p^*.f = f \circ M \tag{15}$$

and thus $(Q', L', c) = ({}^tM\,QM, LM, c)$.

3.5 Other operations

We shall approximate other operations by releasing all constraints on the variables affected by them: forgetting variable in set V is achieved as follows:

- $q'_{i,j} = q_{i,j}$ if $i \notin V$ and $j \notin V$, $q'_{i,j} = 0$ otherwise;
- $L'_i = L_i$ if $i \notin V$, $L_i = 0$.

It is quite obvious that if $f : E \mapsto E$ leaves all coordinates outside of V intact, $G_{Q,L,q_0} \circ f \leq G_{Q',L',q_0}$ point-wise.

4 Mathematical facts

We shall see now a few mathematical points on second-degree polynomials and positive quadratic forms.

4.1 Parabolas

Let $P_1(x) = a_1x^2 + b_1x + c_1$ and $P_2(x) = a_2x^2 + b_2x + c_2$. Let us find a quadric polynomial $P_3(x) = a_3x^2 + b_3x + c_3$ less than P_1 and P_2. Obviously, a_3 must be less than both a_1 and a_2, else P_3 is above P_1 or P_2 near $\pm\infty$.

Let us first suppose that neither $P_1 \leq P_2$ nor $P_2 \leq P_1$ pointwise, since those cases have an obvious solution. Let us remark that this condition holds if and only if P_1 and P_2 intersect, that is, when $\mathrm{discr}(P_1 - P_2) > 0$ — $\mathrm{discr}(ax^2+bx+c)$ is the discriminant $b^2 - 4ac$.

Let us note that there is in general no "greatest lower bound" among quadratic polynomials. The first choice, an arbitrary one, will thus be of any positive a_3 less than a_1 and a_2.

We choose P_3 to be tangent to both P_1 and P_2. This means that $P_3 - P_1$ and $P_3' - P_1'$ have a common root (resp. for $P_3 - P_2$ and $P_3' - P_2'$). This is equivalent to $P_3 - P_1$ and $P_2 - P_1$ each having a double root. That property is ensured by the conditions on the discriminants of the polynomials: $\mathrm{discr}(P_3 - P_1) = 0$ and $\mathrm{discr}(P_3 - P_2) = 0$, that is:

$$(b_3 - b_1)^2 = 4(a_3 - a_1)(c_3 - c_1) \tag{16}$$

$$(b_3 - b_2)^2 = 4(a_3 - a_2)(c_3 - c_2) \tag{17}$$

Let us suppose for now that $a_1 > a_2$. Solving this 2-unknown, 2-equation system yields:

$$b_3 = \frac{-a_2b_1 + a_3(b_1 - b_2) + a_1b_2 \pm \sqrt{\Delta}}{a_1 - a_2} \tag{18}$$

$$\Delta = (a_1 - a_3)(-a_2 + a_3)(-(b_1 - b_2)^2 + 4(a_1 - a_2)(c_1 - c_2)) \tag{19}$$

There are two solutions for this system, which means that for any choice of a_3, there are two polynomials P_3 corresponding to two parabolas tangent to both P_1 and P_2. We wish P_3 to be pointwise less than P_1 and P_2, but we would prefer it not to be too "low"; for this reason, between the two choices, we choose the one for which $\inf P_3 = -\frac{b_3}{2a_3}$ is maximal and thus b_3 is minimal. Since $a_1 > a_2$, this means that we choose

$$b_3 = \frac{-a_2b_1 + a_3(b_1 - b_2) + a_1b_2 - \sqrt{\Delta}}{a_1 - a_2} \tag{20}$$

$$c_3 = c_1 + \frac{(b_3 - b_1)^2}{4(a_3 - a_1)} \tag{21}$$

The case $a_1 < a_2$ is treated *mutatis mutandis*.

Let us now treat $a_1 = a_2$, which is a degenerate case.

$$b_3 = \frac{b1 + b2}{2} - 2\frac{(a1 - a3)(c1 - c2)}{b1 - b2} \tag{22}$$

$$c_3 = c_1 + \frac{(b_3 - b_1)^2}{4(a_3 - a_1)} \tag{23}$$

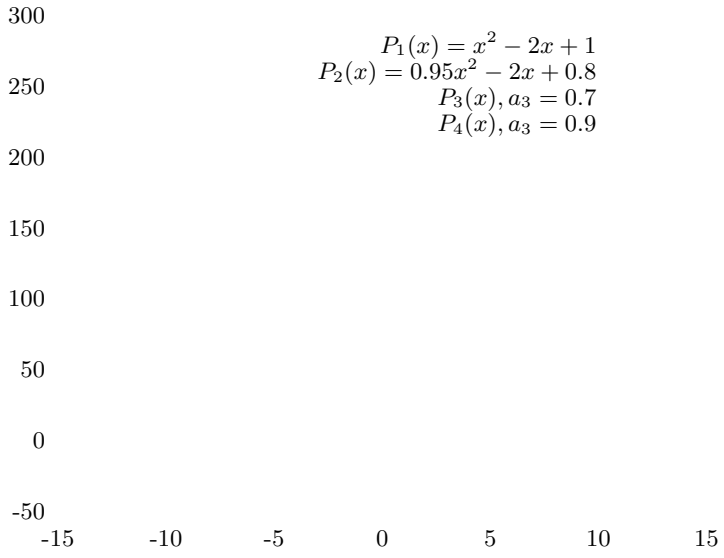

Fig. 2. An example of common lower bounds in quadratic polynomials (P_4 and P_3 for P_1 and P_2).

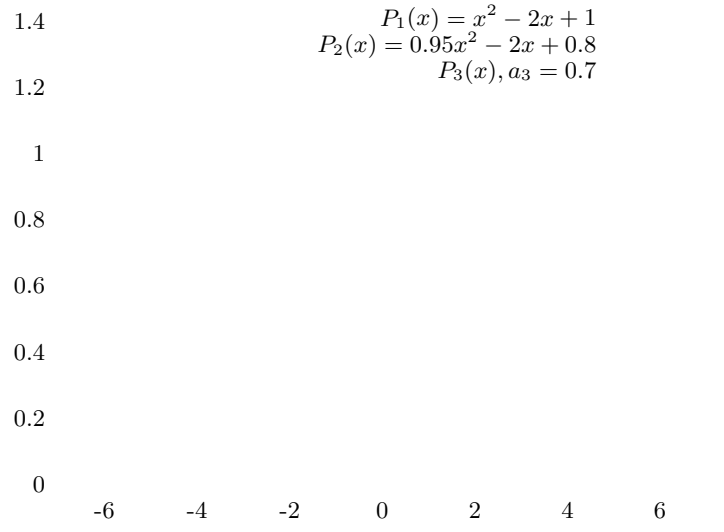

Fig. 3. The Gaussians corresponding to the preceding figure. Note that while the central part is grossly overestimated (and truncated in the figure), the tails are finely approximated.

4.2 Positive quadratic forms

Extended gaussian distributions are defined using positive quadratic forms. An eigenvalue of zero along an axis indicates that there is no Gaussian constraint along that axis; this is the case, for instance, if this axis corresponds to a variable chosen according to a non-Gaussian distribution.

Basic facts

Definition 2 ((Positive) quadratic forms). *A* quadratic form Q *on a real vector space E is defined by a bilinear symmetric function Q^* (called its* polar form*) from $E \times E$ to $\mathbb{R}$. We note $Q(x) = Q^*(x, x)$. A* positive quadratic form Q *is such that for all x in E, $Q(x) \geq 0$.*

Lemma 1. *Let E be a vector space. Let Q be a quadratic form over E. Let F be a finite dimensional subspace of E so that $F \cap Iso\ Q = \{0\}$. Then F has an orthonormal basis with respect to Q.*

Common diagonalization We shall often have to consider two different quadratic forms at the same time. It is then very useful to consider both of them in the same orthogonal basis. The following theorem guarantees that it is possible provided at least one of the quadratic forms is positive definite (or negative definite) [2, th. III.3.1]:

Theorem 1. *Let E be a finite dimensional vector space. Let Q_1 be a quadratic form over E and Q_2 be a positive definite (or negative definite) quadratic form over E. Then there exists a base where both Q_1 and Q_2 are diagonal.*

We shall suppose that we have implementations of certain numerical algorithms (algorithms 1, 2, 3). Those are available in the general literature as well as free and commercial software [6].

Algorithm 1 QuadDiag0, diagonalize a symmetric matrix in an orthonormal basis

Require: M a symmetric matrix
Ensure: $[P, D]$ where D is a diagonal matrix and P is an orthogonal matrix such that $M = PDP^{-1}$.

Algorithm 2 Orth, get an orthnormal basis of the image space of a matrix

Require: M a matrix
Ensure: B such that Im $B =$ Im M and its columns form an orthonormal free family

Algorithm 3 NULL, get an orthnormal basis of the null space of a matrix

Require: M a matrix
Ensure: B such that Im $B = \ker M$ its columns form an orthonormal free family

Algorithm 4 QUADDIAG1, common diagonalization of a quadratic form and a positive definite quadratic form

Require: $[Q_1, Q_2]$ where Q_1 a symmetric matrix, Q_2 a positive definite symmetric matrix
Ensure: $[P, I, d, D_1]$ where P is an invertible matrix, I its inverse, $d = \det P$, D_1 is a diagonal matrices such that $Q_1 = {}^tI\ D_1 I$ and $Q_2 = {}^tI\ I$
$[P_2, D_2] \leftarrow$ QUADDIAG0$[Q_2]$
$Z \leftarrow D_2^{-1/2}$
$H \leftarrow P_2 Z$
$G \leftarrow {}^tH\ Q_2 H$
$[P_1, D_1] \leftarrow$ QUADDIAG0$[G]$
$I \leftarrow {}^tP_1\ \sqrt{D_2}\ {}^tP_2$
$P \leftarrow P_2 Z P_1$
$d \leftarrow \det Z$

Unfortunately, in our case, we have to handle quadratic forms that have isotropic vectors. On the other hand, we only consider positive forms, and we thus have a theorem:

Theorem 2. *Let E be a finite dimensional vector space. Let n be* $\dim E$*. Let Q_1 and Q_2 be two positive quadratic forms over E. Then there exists a base $(e_i)_{1\leq i\leq n}$ where both Q_1 and Q_2 are diagonal, that is, there exist two families real numbers $(\lambda_i)_{1\leq i\leq n}$ and $(\mu_i)_{1\leq i\leq n}$ so that*

$$Q_1\left(\sum_i \alpha_i e_i\right) = \sum_i \lambda_i \alpha_i^2 \tag{24}$$

$$Q_2\left(\sum_i \alpha_i e_i\right) = \sum_i \mu_i \alpha_i^2. \tag{25}$$

Let us now develop an effective algorithm for this theorem (Alg. 5). For effectiveness reasons, we choose F to be the orthogonal of $\ker Q_1 \cap \ker Q_2$ for the canonic dot product. Since $\ker Q_1 \cap \ker Q_2^{\perp} =$ Im $Q_1 +$ Im Q_2 we obtain an orthonormal basis of F by orthogonalizing a generating family of Im Q_1 (the columns of Q_1), extending that basis to an orthonormal basis of Im $Q_1 +$ Im Q_2 using a generating family of Im Q_2 (the columns of Q_2). We then have an orthonormal basis of F, which can be extended to a basis B of $\mathbb{R}^n$ using a generating family of $\mathbb{R}$ (the canonical basis).

We consider both quadratic forms Q_1 and Q_2 on that basis B. Their matrices are of the form

$$Q_i = {}^tB\ Q_i B = \begin{pmatrix} Q_i' & 0 \\ 0 & 0 \end{pmatrix} \tag{26}$$

where Q'_1 and Q'_2 are square matrices of size $\dim F$. We diagonalize Q'_1 with respect to the definite positive matrix $Q'_1 + Q'_2$ and output the results with respect to the right bases.

Algorithm 5 QUADDIAG2, common diagonalization of two positive quadratic forms

Require: Q_1 and Q_2 two positive symmetric matrices
Ensure: $[P, I, d, D_1, D_2]$ where P is an invertible matrix, I its inverse, $d = \det P$, D_1 and D_2 two diagonal matrices such that $Q_1 = {}^tI\, D_1 I$ and $Q_2 = {}^tI\, D_2 I$
$F \leftarrow \textsc{Orth}\left[\begin{pmatrix} Q1 & Q2 \end{pmatrix}\right]$
$K \leftarrow \textsc{Null}\left[\begin{pmatrix} Q1 \\ Q2 \end{pmatrix}\right]$
$[P', I', d, D'_1] \leftarrow \textsc{QuadDiag1}({}^tF\, Q_1 F, {}^tF\, (Q_1 + Q_2) F)$
$D_1 \leftarrow \begin{pmatrix} D'_1 & 0 \\ 0 & 0 \end{pmatrix}$
$D_2 \leftarrow \begin{pmatrix} 1 - D'_1 & 0 \\ 0 & 0 \end{pmatrix}$
$P \leftarrow \begin{pmatrix} FP' & K \end{pmatrix}$
$I \leftarrow \begin{pmatrix} I' {}^tF \\ {}^tK \end{pmatrix}$

5 Conclusions

We presented an abstract domain for the backwards abstract interpretation of probabilistic programs, with a view to representing exactly the properties of programs using normally distributed generators.

As shown in Fig. 3, this analysis yields coarse results in the center of the functions; on the other hand, it leads to very precise results in the tails of the distribution. It therefore seems desirable to use it as a way to bound the influence of the tails of the random generators while using other methods, including abstract Monte-Carlo [12], for the center of the distribution.

The main problem with this domain is that it does not interact well with precise bounds, obtained for instance with a test with respect to an interval. A possible direction of research is an abstract domain covering both precise bounds and Gaussian bounds.

An application of this Gaussian analysis could be the study of the propagation of computational inaccuracies introduced by floating-point arithmetics, modeled by random choices.[1] We hope to contribute to that recently opened field [16] of abstract interpretation applied to round-off errors.

References

1. E. Anderson, Z. Bai, C. Bischof, S. Blackfordand J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK*

[1] This idea of modeling inaccuracies by random choices is the basis of the CESTAC method [17].

Users' Guide. SIAM, third edition, 1999. On-line extra documentation at `http://www.netlib.org/lapack/`.
2. J.M. Arnaudiès and H. Fraysse. *Cours de mathématiques, 4 : Algèbre bilinéaire et géométrie*. Dunod Université, 1990.
3. Philippe G. Ciarlet. *Introduction à l'analyse numérique matricielle et à l'optimisation*. Masson, Paris, 1982.
4. Patrick Cousot and Radhia Cousot. Abstract interpretation and application to logic programs. *J. Logic Prog.*, 2-3(13):103–179, 1992.
5. Luca de Alfaro, Marta Kwiatkowska, Gethin Norman, David Parker, and Roberto Segala. Symbolic model checking of probabilistic processes using MTBDDs and the kronecker representation. In *TACAS'2000*, volume 1785 of *Lecture Notes in Computer Science*. Springer-Verlag, January 2000.
6. Free Software Foundation. *GNU Octave: A high-level interactive language for numerical computations*.
   `http://www.octave.org/doc/octave_toc.html`
7. Michael Huth and Marta Kwiatkowska. On probabilistic model checking. Technical Report CSR-96-15, University of Birmingham, School of Computer Science, August 1996.
   `ftp://ftp.cs.bham.ac.uk/pub/tech-reports/1996/CSR-96-15.ps.gz`
8. D. Kozen. Semantics of probabilistic programs. In *20th Annual Symposium on Foundations of Computer Science*, pages 101–114, Long Beach, Ca., USA, October 1979. IEEE Computer Society Press.
9. D. Kozen. Semantics of probabilistic programs. *Journal of Computer and System Sciences*, 22(3):328–350, 1981.
10. Marta Z. Kwiatkowska, Gethin Norman, Roberto Segala, and Jeremy Sproston. Verifying quantitative properties of continuous probabilistic timed automata. Technical Report CSR-00-6, University of Birmingham, School of Computer Science, March 2000.
    `ftp://ftp.cs.bham.ac.uk/pub/tech-reports/2000/CSR-00-06.ps.gz`
11. David Monniaux. Abstract interpretation of probabilistic semantics. In *Seventh International Static Analysis Symposium (SAS'00)*, number 1824 in Lecture Notes in Computer Science, pages 322–339. Springer-Verlag, 2000. Extended version on the author's web site.
12. David Monniaux. An abstract Monte-Carlo method for the analysis of probabilistic programs (extended abstract). In *28th Symposium on Principles of Programming Languages (POPL '01)*, pages 93–101. Association for Computer Machinery, 2001.
13. David Monniaux. *Analyse de programmes probabilistes par interprétation abstraite*. Thèse de doctorat, Université Paris IX Dauphine, 2001. Résumé étendu en français. Contents in English.
14. David Monniaux. Backwards abstract interpretation of probabilistic programs. In *European Symposium on Programming Languages and Systems (ESOP '01)*, number 2028 in Lecture Notes in Computer Science, pages 367–382. Springer-Verlag, 2001.
15. Martin L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. Wiley series in probability and mathematical statistics. John Wiley & Sons, 1994.
16. Éric Goubault. Static analyses of floating-point operations. In *Static Analysis (SAS '01)*, Lecture Notes in Computer Science. Springer-Verlag, July 2001.
17. J. Vignes and R. Alt. An efficient stochastic method for round-off error analysis. In *Accurate scientific computations (Bad Neuenahr, 1985)*, pages 183–205. Springer, Berlin, 1986.