Weighted Model Integration with Orthogonal Transformations

David Merrell, Aws Albarghouthi, and Loris D'Antoni

Weighted Model Integration

Weighted Model Integration (WMI) is an **exact probabilistic inference method** for probabilistic models. Inference in **probabilistic programs** can be reduced to WMI.

Probabilistic Program

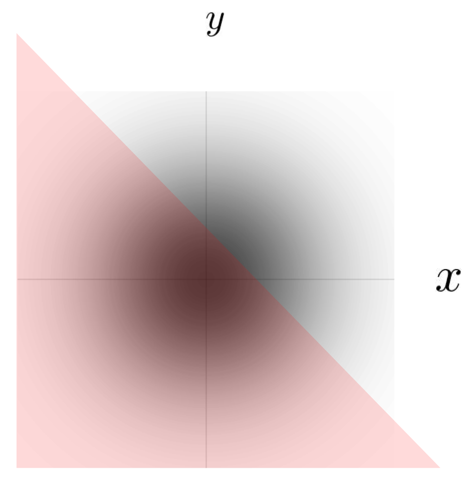
```
x = Normal(0,9);
y = Normal(0,9);
z = x + y;

Pr{z < 1}
```

First-order Formula

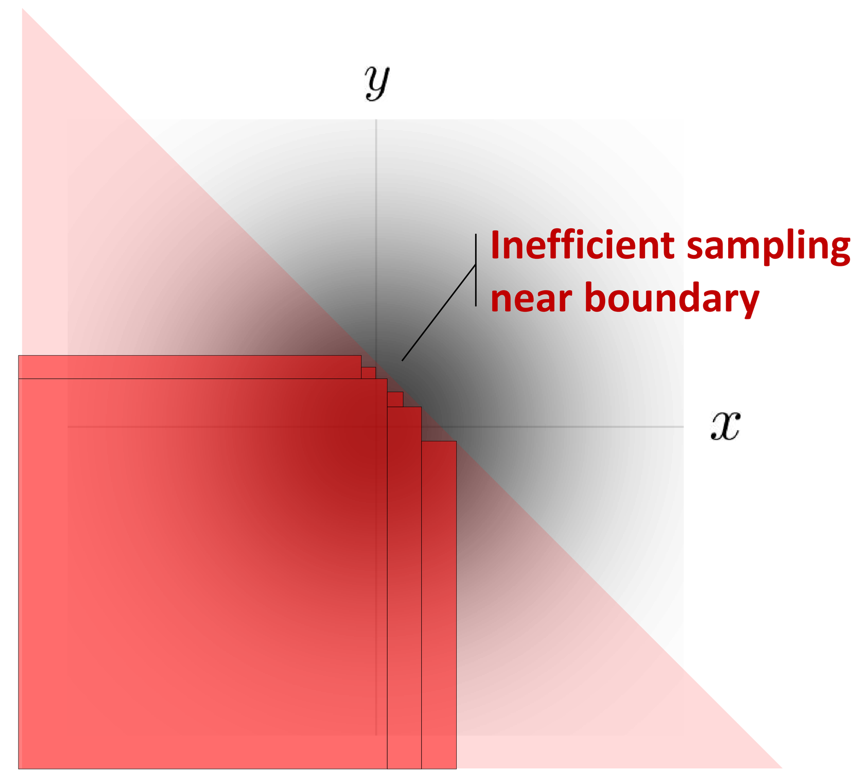
$$\varphi \equiv \begin{array}{c} \exists z.\ z = x + y \\ \wedge\ z < 1 \end{array}$$

$$x, y \sim N(0, 9)$$

Integration Problem



Rectangular Decomposition Method



SMT Formula for Rectangular Samples

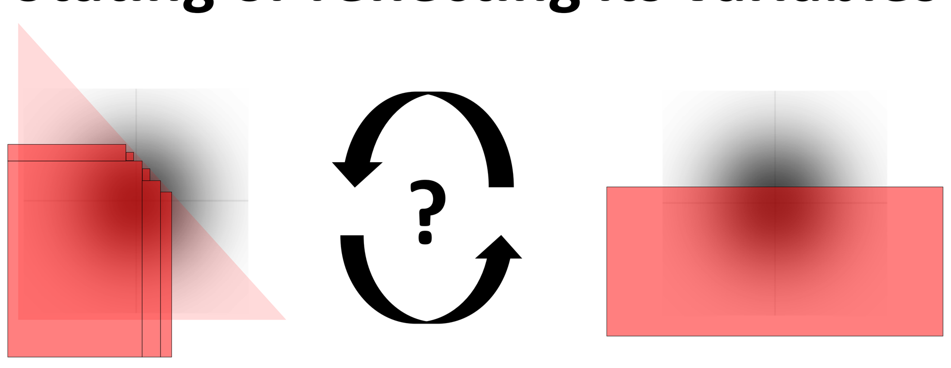
$$H_\varphi \equiv \left(\bigwedge_{x \in \mathcal{X}_\varphi} l_x \leq u_x \right) \wedge \forall \mathcal{X}_\varphi . \left(\bigwedge_{x \in \mathcal{X}_\varphi} l_x \leq x \leq u_x \right) \Rightarrow \varphi$$

Rectangular Decomposition Methods use an **SMT solver** to sample axis-aligned rectangles from the region of integration. The probabilities of these rectangles are easy to compute.

Rectangular Decomposition can suffer from inefficient sampling.

Orthogonal Transformations to the Rescue!

The Big Idea:
Can we improve a formula's sampling efficiency by **rotating or reflecting its variables**?



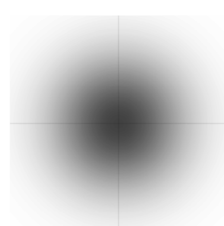
Some questions need to be answered…

Q: *What distributions will this work for?*

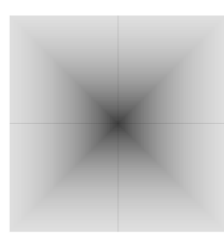
A: Skitovitch-Darmois Theorem $\Longrightarrow$ **Only Gaussians.**

$$X_i \text{ independent} \qquad \alpha_i, \beta_i \neq 0 \qquad i \in \{1 \ldots k\}$$

$$\left. \begin{array}{l} L_1 = \sum_{i=1}^{k} \alpha_i X_i \\ L_2 = \sum_{i=1}^{k} \beta_i X_i \end{array} \right\} \text{independent} \Longrightarrow X_i \sim N(\mu_i, \sigma_i^2)$$



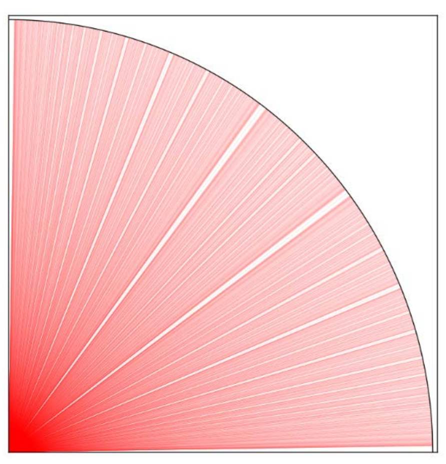
Rotated Gaussians still have a product distribution.

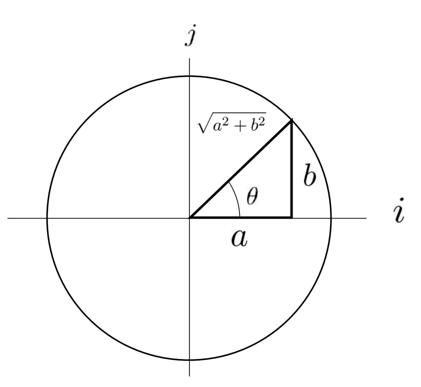


E.g., rotated Laplacians no longer have a product distribution.

Q: *Rotations may introduce irrational numbers, which are bad for SMT solvers. How do we avoid that?*

A: Use **Pythagorean triples** to generate **rational Givens rotations**:





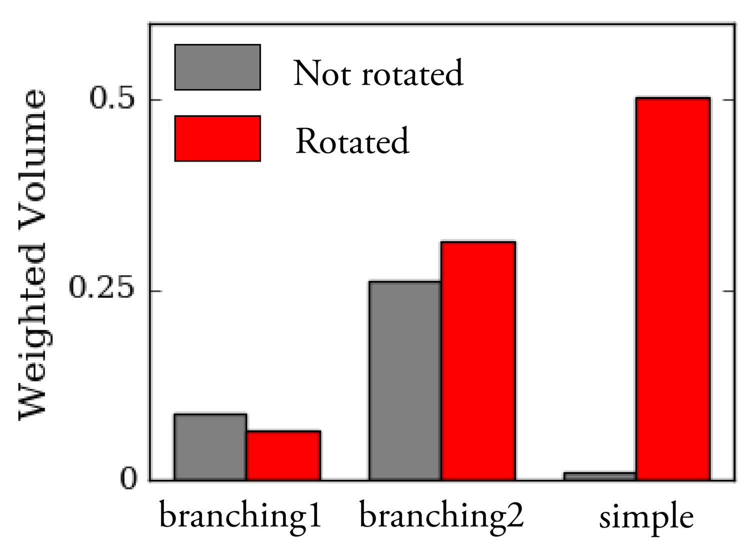
$$G(i,j) = \frac{1}{\sqrt{a^2+b^2}} \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & a & \cdots & -b & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & b & \cdots & a & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$

We compose multiple Givens rotations to obtain a rotation matrix that (approximately) aligns the formula's faces with the axes.

Our method is related to **QR-factorization by Givens rotations.**

Q: *Does this actually improve efficiency?*

A: Yes. Rotations improved sampling efficiency for 10 of 12 probabilistic program benchmarks. In some programs, rotations made little difference either way; in others it led to dramatic improvements. For example:



Probability captured in 100s of rectangular sampling, for three example programs, with and without rotation.

This work is generously funded by NSF Grant #1704117: **Formal Methods for Program Fairness.**
See *Quantifying Program Bias*, Albarghouthi et al, OOPSLA 2017 **(FairSquare)** for related work.