Amortized Rejection Sampling in Universal Probabilistic Programming

Saeid Naderiparizi[1], Adam Ścibior[1], Andreas Munk[1], Mehrdad Ghadiri[2], Atılım Güneş Baydin[3], Bradley Gram-Hansen[3], Christian Schroeder de Witt[3], Robert Zinkov[3], Philip Torr[3], Tom Rainforth[3], Yee Whye Teh[3], Frank Wood[1,4,5]

[1]University of British Columbia, [2]Georgia Institute of Technology, [3]University of Oxford, [4]MILA, [5]CIFAR AI Chair

AISTATS 2022

Introduction

- The goal of probabilistic programming is to separate the process of modeling and inference, and allow automatic inferece.
- Universal probabilistic programming (also known as "Turing complete" or "higher order" probabilistic programming) supports programs with an unbounded number of random variables.
- Such flexibility makes probabilsitic programs easily accessible to a broad audience. However, it introduces new challenges to the design of reliable inference algorithms.
- One of these challenges that we focus on in this paper is when the program contains rejection sampling loops.

Problem formulation

We present our method on the example program (a).

In this program:

- latent variables: x and z^ks.
- observed variable: y.

```
x ~ p(x)
for k ∈ ℕ⁺ do
    z^k ~ p(z|x)
    if c(x, z^k) then
        z = z^k
        break
    observe(y, p(y|z, x))
```

(a) Original program

Problem formulation

$x \sim p(x)$

for $k \in \mathbb{N}^+$ **do**

  $z^k \sim p(z|x)$

  if $c(x, z^k)$ **then**

    $z = z^k$

    break

  `observe`$(y,\ p(y|z,x))$

(a) Original program

$x \sim q(x|y)$

$w \leftarrow \frac{p(x)}{q(x|y)}$

for $k \in \mathbb{N}^+$ **do**

  $z^k \sim q(z|x,y)$

  $w^k \leftarrow \frac{p(z^k|x)}{q(z^k|x,y)}$

  $w \leftarrow w\, w^k$

  if $c(x, z^k)$ **then**

    $z = z^k$

    break

$w_{\text{IC}} \leftarrow wp(y|z,x)$

(b) Inference Compilation

Problem formulation

$$w_{\text{IC}} = \frac{p(x)}{q(x|y)} p(y|x,z) \prod_{k=1}^{L} \frac{p(z^k|x)}{q(z^k|x,y)}$$

where L is a random variable denoting the number of rejection sampling iterations until an acceptance.

(a) Original program	(b) Inference Compilation
$x \sim p(x)$	$x \sim q(x\|y)$
	$w \leftarrow \frac{p(x)}{q(x\|y)}$
for $k \in \mathbb{N}^+$ **do**	**for** $k \in \mathbb{N}^+$ **do**
$\quad z^k \sim p(z\|x)$	$\quad z^k \sim q(z\|x,y)$
	$\quad w^k \leftarrow \frac{p(z^k\|x)}{q(z^k\|x,y)}$
	$\quad w \leftarrow w\, w^k$
$\quad$ **if** $c(x,z^k)$ **then**	$\quad$ **if** $c(x,z^k)$ **then**
$\quad\quad z = z^k$	$\quad\quad z = z^k$
$\quad\quad$ **break**	$\quad\quad$ **break**
$\quad$ `observe`$(y,\ p(y\|z,x))$	$w_{\text{IC}} \leftarrow wp(y\|z,x)$

Problem formulation

$$w_{\text{IC}} = \frac{p(x)}{q(x|y)} p(y|x,z) \prod_{k=1}^{L} \frac{p(z^k|x)}{q(z^k|x,y)}$$

Theorem

Under some regulatory conditions, if the following condition holds with positive probability under $x \sim q(x|y)$

$$\mathbb{E}_{z\sim q(z|x,y)}\left[\frac{p(z|x)^2}{q(z|x,y)^2}(1-p(A|x,z))\right] \geq 1$$

where A is the event that c is satisfied, then the variance of w_{IC} is infinite.

$x \sim p(x)$

for $k \in \mathbb{N}^+$ **do**
  $z^k \sim p(z|x)$
  if $c(x, z^k)$ **then**
    $z = z^k$
    break
  `observe`$(y,\ p(y|z,x))$

(a) Original program

$x \sim q(x|y)$
$w \leftarrow \frac{p(x)}{q(x|y)}$
for $k \in \mathbb{N}^+$ **do**
  $z^k \sim q(z|x,y)$
  $w^k \leftarrow \frac{p(z^k|x)}{q(z^k|x,y)}$
  $w \leftarrow w\, w^k$
  if $c(x, z^k)$ **then**
    $z = z^k$
    break
$w_{\text{IC}} \leftarrow wp(y|z,x)$

(b) Inference Compilation

Approach

$$w_{\text{IC}} = \frac{p(x)}{q(x|y)} p(y|x,z) \prod_{k=1}^{L} \frac{p(z^k|x)}{q(z^k|x,y)}$$

Theorem

Under some regulatory conditions, if the following condition holds with positive probability under $x \sim q(x|y)$

$$\mathbb{E}_{z\sim q(z|x,y)}\left[\frac{p(z|x)^2}{q(z|x,y)^2}(1-p(A|x,z))\right] \geq 1$$

where A is the event that c is satisfied, then the variance of w_{IC} is infinite.

```
x ~ p(x)

for k ∈ N+ do
    z^k ~ p(z|x)


    if c(x, z^k) then
        z = z^k
        break
    observe(y, p(y|z, x))
```

(a) Original program

```
x ~ q(x|y)
w ← p(x)/q(x|y)
for k ∈ N+ do
    z^k ~ q(z|x, y)
    w^k ← p(z^k|x)/q(z^k|x, y)
    w ← w w^k
    if c(x, z^k) then
        z = z^k
        break
w_IC ← w p(y|z, x)
```

(b) Inference Compilation

```
x ~ p(x)

z ~ p(z|x, c(x, z))

observe(y, p(y|z, x))
```

(c) Collapsed program

Approach

$$w_{\mathrm{C}} = \frac{p(x)}{q(x|y)} \frac{p(z|x, A)}{q(z|x, y, A)} p(y|x, z)$$

where A is the event that c is satisfied.

$x \sim p(x)$

for $k \in \mathbb{N}^+$ **do**
  $z^k \sim p(z|x)$
  if $c(x, z^k)$ **then**
    $z = z^k$
    break
  `observe`$(y,\ p(y|z, x))$

(a) Original program

$x \sim q(x|y)$
$w \leftarrow \frac{p(x)}{q(x|y)}$
for $k \in \mathbb{N}^+$ **do**
  $z^k \sim q(z|x, y)$
  $w^k \leftarrow \frac{p(z^k|x)}{q(z^k|x,y)}$
  $w \leftarrow w\, w^k$
  if $c(x, z^k)$ **then**
    $z = z^k$
    break
$w_{\mathrm{IC}} \leftarrow w p(y|z, x)$

(b) Inference Compilation

$x \sim p(x)$

$z \sim p(z|x, c(x, z))$

`observe`$(y,\ p(y|z, x))$

(c) Collapsed program

$x \sim q(x|y)$
$w \leftarrow \frac{p(x)}{q(x|y)}$
$z \sim q(z|x, y, c(x, z))$
$w \leftarrow w\, \frac{p(z|x,c(x,z))}{q(z|x,y,c(x,z))}$
$w_{\mathrm{C}} \leftarrow w\, p(y|z, x)$

(d) Our IS estimator

Approach

$$w_{\mathrm{C}} = \frac{p(x)}{q(x|y)} \frac{p(z|x, A)}{q(z|x, y, A)} p(y|x, z)$$
$$= \frac{p(x)}{q(x|y)} \frac{p(z|x)}{q(z|x, y)} \frac{q(A|x, y)}{p(A|x)} p(y|x, z)$$

```
x ~ p(x)

for k ∈ N+ do
  z^k ~ p(z|x)


  if c(x, z^k) then
    z = z^k
    break
  observe(y, p(y|z, x))
```

(a) Original program

```
x ~ q(x|y)
w ← p(x) / q(x|y)
for k ∈ N+ do
  z^k ~ q(z|x, y)
  w^k ← p(z^k|x) / q(z^k|x, y)
  w ← w w^k
  if c(x, z^k) then
    z = z^k
    break
w_IC ← w p(y|z, x)
```

(b) Inference Compilation

```
x ~ p(x)

z ~ p(z|x, c(x, z))

observe(y, p(y|z, x))
```

(c) Collapsed program

```
x ~ q(x|y)
w ← p(x) / q(x|y)
z ~ q(z|x, y, c(x, z))
w ← w p(z|x, c(x, z)) / q(z|x, y, c(x, z))
w_C ← w p(y|z, x)
```

(d) Our IS estimator

Approach

$$w_{\mathrm{C}} = \frac{p(x)}{q(x|y)} \frac{p(z|x,A)}{q(z|x,y,A)} p(y|x,z)$$
$$= \frac{p(x)}{q(x|y)} \frac{p(z|x)}{q(z|x,y)} \frac{q(A|x,y)}{p(A|x)} p(y|x,z)$$

- Amortized Rejection Sampling (ARS) gets an unbiased estimate of $\frac{q(A|x,y)}{p(A|x)}$ by multiplying estimates of $q(A|x,y)$ and $\frac{1}{p(A|x)}$ obtained by Monte Carlo procedures.
- We prove that ARS does not introduce infinite variance due to rejection loops.
- We implement ARS in PyProb.[3]

$x \sim p(x)$

for $k \in \mathbb{N}^+$ **do**

  $z^k \sim p(z|x)$

  if $c(x, z^k)$ **then**

    $z = z^k$

    break

  observe$(y,\ p(y|z,x))$

(a) Original program

$x \sim q(x|y)$

$w \leftarrow \frac{p(x)}{q(x|y)}$

for $k \in \mathbb{N}^+$ **do**

  $z^k \sim q(z|x,y)$

  $w^k \leftarrow \frac{p(z^k|x)}{q(z^k|x,y)}$

  $w \leftarrow w\, w^k$

  if $c(x, z^k)$ **then**

    $z = z^k$

    break

$w_{\mathrm{IC}} \leftarrow w p(y|z,x)$

(b) Inference Compilation

$x \sim p(x)$

$z \sim p(z|x, c(x,z))$

observe$(y,\ p(y|z,x))$

(c) Collapsed program

$x \sim q(x|y)$

$w \leftarrow \frac{p(x)}{q(x|y)}$

$z \sim q(z|x,y,c(x,z))$

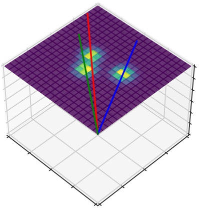
$w \leftarrow w\, \frac{p(z|x,c(x,z))}{q(z|x,y,c(x,z))}$

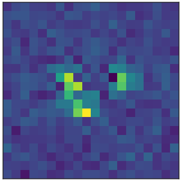
$w_{\mathrm{C}} \leftarrow w\, p(y|z,x)$

(d) Our IS estimator

[1] https://github.com/pyprob/pyprob.

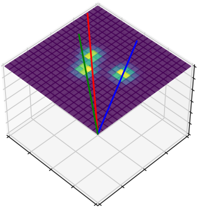
Experiments - Mini-SHERPA

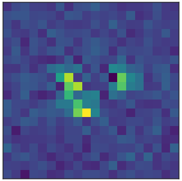




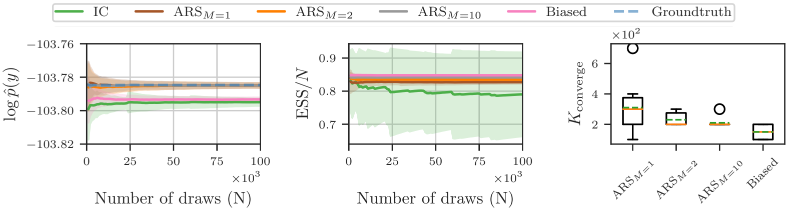
- Mini-SHERPA is an event generator of a simplified model of high-energy reactions of particles.
- It has up to two rejection sampling loops.

Experiments - Mini-SHERPA





- Mini-SHERPA is an event generator of a simplified model of high-energy reactions of particles.
- It has up to two rejection sampling loops.



Experiments - Beta-Bernoulli

for $k \in \mathbb{N}^+$ **do**
    $x_k \sim \text{Uniform}(0, 1)$
    $u_k \sim \text{Uniform}(0, 1)$
    if $c(x_k, u_k)$ **then**
        $x = x_k$
        $u = u_k$
        break
for $i \in [1, n]$ **do**
    `observe`$(\text{True}, \text{Ber}(x))$

Implementation

$x \sim p(x)$
for $k \in \mathbb{N}^+$ **do**
  $z^k \sim p(z|x)$
  if $c(x, z^k)$ **then**
    $z = z^k$
    break
  observe$(y, p(y|z, x))$

Program 1: Original

```
x = sample(P_x)
while True:

    z = sample(P_z(x))
    if c(x, z):

        break
observe(P_y(x,z), y)
return x, z
```

Program 2: Annotated

```
x = sample(P_x)
while True:
    rs_start()
    z = sample(P_z(x))
    if c(x, z):
        rs_end()
        break
observe(P_y(x,z), y)
return x, z
```

Conclusion

- We have demonstrated theoretically and empirically that even simple rejection sampling loops can cause major problems for existing probabilistic programming inference algorithms.
- We have proposed an alternative way to compute importance sampling weights in such programs.
- We have proved that our method is not susceptible to the rejection sampling loop problems.
- We have implemented our method in an existing universal probabilistic programming framework.

The implementation of models used in experiments and ARS in pyprob can be found here:
`https://github.com/plai-group/amortized-rejection-sampling`

- The implementation of models used in experiments and ARS in pyprob can be found here: `https://github.com/plai-group/amortized-rejection-sampling`

Thank you