Amortized Rejection Sampling in Universal Probabilistic Programming

Saeid Naderiparizi, Adam Ścibior, Andreas Munk, Mehrdad Ghdiri, Atılım Güneş Baydin, Bradley Gram-Hansen, Christian Schroeder de Witt, Robert Zinkov, Philip Torr, Tom Rainforth, Yee Whye Teh, Frank Wood

TL;DR

- Rejection sampling is widely used in implementing complex generative models.
- Inference in probabilistic programs including unbounded loops (e.g. rejection sampling) is hard.
- We address the problem of efficient amortized importance-sampling-based inference, in particular Inference Compilation (IC) [4], in such models.
- We show naive application of IC can produce importance weights with unbounded variance.
- We propose Amortized Rejection Sampling (ARS), an importance sampling procedure that produces finite variance weights and unbiased expectations for programs that include rejection sampling loops.
- We implement ARS in pyroprob [1; 2] in a way that requires minimal modifications to user code.

**Original**

\[
x = \text{sample}( P_x )
\text{while True:}
\]
\[
z = \text{sample}( P_z(x) )
\text{if } c(x, z):\]
\[
\text{break}
\]
\[
\text{observe}( P_y(x,z), y )
\]
\[
\text{return } x, z
\]

**Annotated**

\[
x = \text{sample}( P_x )
\text{while True:}
\]
\[
\text{rs start ()}
z = \text{sample}( P_z(x) )
\text{if } c(x, z):\]
\[
\text{rs end ()}
\text{break}
\]
\[
\text{observe}( P_y(x,z), y )
\]
\[
\text{return } x, z
\]

**ARS, M=1**

\[
\text{IC weights:}
\]
\[
u_{IC} = \frac{p(x)}{q(x|y)p(y|x,z)}|L_{IC}^- u^k
\]

**Theorem:** Under some mild conditions if the following holds then the variance of \( u_{IC} \) is infinite.

\[
E_{x \sim p(x),y \sim q(y|x,z)} \left( \frac{p(y|x,z)}{q(y|x,z)} (1 - p(A|x, z)) \right) \geq 1
\]

where \( A \) is the event of \( c(x, z) \) being satisfied.

**Collapsed weights**

\[
u_{C} = \frac{p(x)p(z|x)}{q(x|y)p(y|x,z)}
\]

- \( q(A|x, y) \) is the probability of exiting the rejection sampling loop under the proposal.
- \( p(A|x) \) is the probability of exiting the rejection sampling loop in the original probabilistic program.
- We use Monte Carlo to get unbiased estimates of \( q(A|x, y) \) and \( p(A|x) \).

**Implementation**

We introduce two new functions to tag the beginning and end of rejection sampling loops.

**Original**

\[
x = \text{sample}(P_x)
\text{while True:}
\]
\[
\text{rs start ()}
x = \text{sample}(P_x)
\text{if } c(x, z):\]
\[
\text{rs end ()}
\text{break}
\]
\[
\text{observe}(P_y(x,z), y)
\]
\[
\text{return } x, z
\]

**Annotated**

\[
x = \text{sample}(P_x)
\text{while True:}
\]
\[
\text{rs start ()}
x = \text{sample}(P_x)
\text{if } c(x, z):\]
\[
\text{rs end ()}
\text{break}
\]
\[
\text{observe}(P_y(x,z), y)
\]
\[
\text{return } x, z
\]

**Mini-SHERPA**

Mini-SHERPA is a simplified model of high-energy reactions of particles [3]. It uses rejection sampling extensively to simulate a particle decay event and the energy deposited by the resulting particles in a simplified detector.

**Algorithm**

1. \( x \sim q(x|y) \)
2. \( w \leftarrow \frac{q(x|y)}{p(x)} \)
3. for \( k \in N^0 \) do
4. \( z \sim q(z|x) \)
5. \( w' \leftarrow \frac{w}{q(z|x)} \)
6. if \( c(x, z) \) then
7. \( z \leftarrow z' \)
8. \( w' \leftarrow w' \)
9. \( K \leftarrow 0 \)
10. for \( i \in N^0 \) do
11. \( z_i \sim q(z|x) \)
12. \( K \leftarrow K + z_i \)

**References**