Adapting Probabilistic Programming Systems to **Estimate Expectations** Efficiently

Expectation Programming

MOTIVATION

Most statistical workflows require calculating an expectation. Standard probabilistic programming systems (PPSs) focus on automating the computation of the posterior $p(x|y)$ and then use Monte Carlo methods to estimate an expectation $\mathbb{E}_{p(x|y)}[f(x)]$. If the target function $f(x)$ is known ahead of time, this pipeline is inefficient . We introduce the concept of an *Expectation Programming Framework (EPF)*. Whereas PPSs can be viewed as tools for approximating conditional distributions, the aim of the inference engine in an EPF is to directly *estimate expectations*.

EXPECTATION PROGRAMMING IN TURING

- We introduce a specific implementation of an EPF, called **EPT** (Expectation Programming in Turing), built upon *Turing* [2]
- In EPT, **programs define expectations**
- EPT takes as input a Turing-style program and uses program transformations to create a new set of three valid Turing programs to **construct target-aware estimators**
- We can repurpose any native Turing inference algorithm that provides a marginal likelihood estimate into a target-aware inference strategy
- We show that EPT provides **significant empirical gains** in practice

BACKGROUND

The recently proposed Target-Aware Bayesian Inference (TABI) framework of [1] provides a means of creating a target-aware estimator by breaking the expectation into three parts

$$\mathbb{E}_{p(x|y)}[f(x)] = \frac{Z_1^+ - Z_1^-}{Z_2}$$

where

$$Z_1^+ = \int p(x,y) max(f(x),0) dx,$$

$$Z_1^- = \int p(x,y) max(-f(x),0) dx$$

$$Z_2 = \int p(x,y) dx$$

```
@expectation function expt_prog(y)
    x ~ Normal(0, 1)
    y ~ Normal(x, 1)
    return x^3
end
```

```
@model function gamma1_plus(y)
    x ~ Normal(0, 1)
    y ~ Normal(x, 1)
    tmp = x^3
    @addlogprob!(log(max(tmp, 0)))
    return tmp
end
```

```
@model function gamma1_minus(y)
    x ~ Normal(0, 1)
    y ~ Normal(x, 1)
    tmp = x^3
    @addlogprob!(log(-min(tmp, 0)))
    return tmp
end
```

```
@model function gamma2(y)
    x ~ Normal(0, 1)
    y ~ Normal(x, 1)
    return x^3
end
```

Figure 1: An EPT program (left) gets transformed into three valid Turing programs (right). The Turing programs can be used to estimate the expectation defined by the input program in a target-aware manner.

STATISTICAL VALIDITY

We provide a proof of the statistical correctness of the EPT approach.

Theorem 1. *Let $\mathcal{E}$ be a valid program in EPT with unnormalized density $\gamma(x_{1:n})$ and reference measure $\mu(x_{1:n})$, defined on possible traces $x_{1:n} \in \mathcal{X}$, and return value $F = f(x_{1:n})$. Then $\gamma_1^+(x_{1:n}) \coloneqq \gamma(x_{1:n})\max(0, f(x_{1:n}))$, $\gamma_1^-(x_{1:n}) \coloneqq \gamma(x_{1:n})\max(0, -f(x_{1:n}))$, and $\gamma_2(x_{1:n}) \coloneqq \gamma(x_{1:n})$ are all valid unnormalized probabilistic program densities. Further, if $\{\hat{Z}_1^+\}_m, \{\hat{Z}_1^-\}_m, \{\hat{Z}_2\}_m$ are sequences of estimators for $m \in \mathbb{N}^+$ such that*

$$\{\hat{Z}_1^\pm\}_m \xrightarrow{p} \int_{\mathcal{X}} \gamma_1^\pm(x_{1:n}) d\mu(x_{1:n}),$$

$$\{\hat{Z}_2\}_m \xrightarrow{p} \int_{\mathcal{X}} \gamma_2(x_{1:n}) d\mu(x_{1:n})$$

Where $\xrightarrow{p}$ means convergence in probability as $m \to \infty$, then

$$\frac{\left(\{\hat{Z}_1^+\}_m - \{\hat{Z}_1^-\}_m\right)}{\{\hat{Z}_2\}_m} \xrightarrow{p} \mathbb{E}[F].$$

EXPERIMENTS

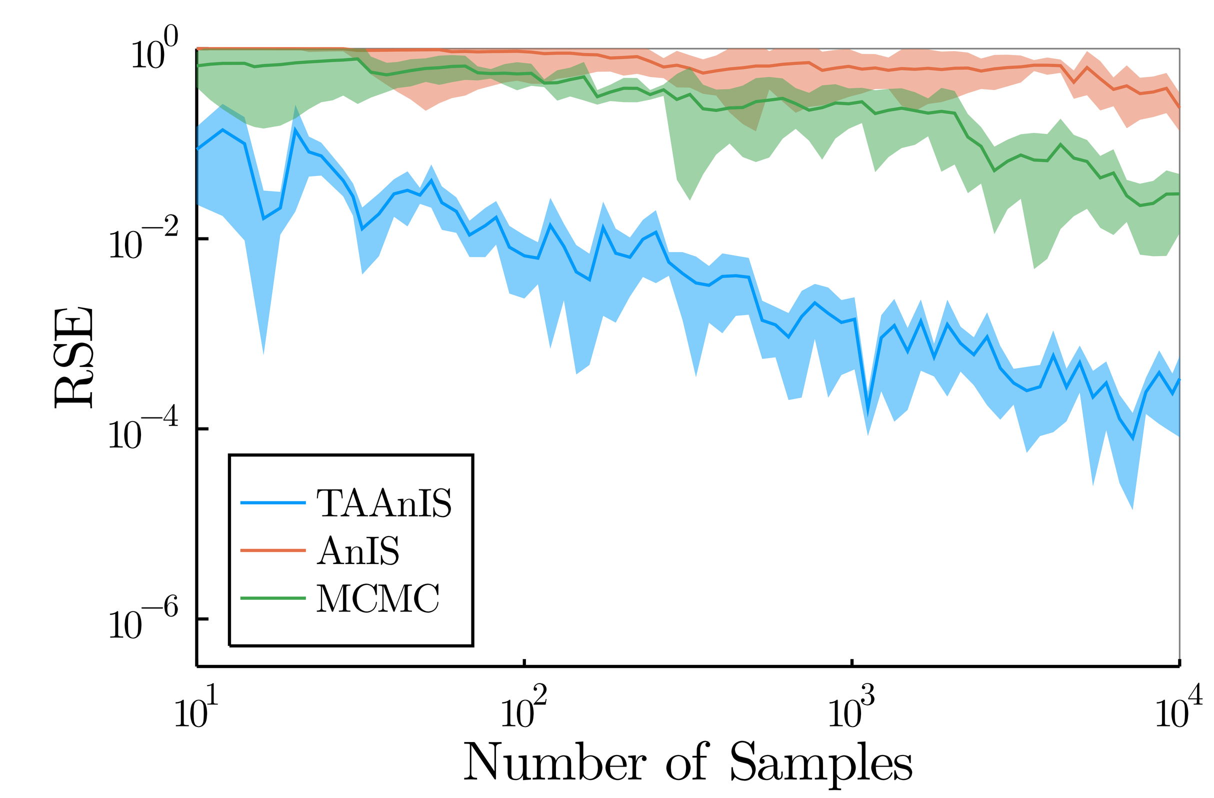

Figure 2: Relative Squared Error (RSE) for estimating the posterior predictive density of a Gaussian model. The target-aware estimator (TAAnIS) significantly outperforms the two baselines.

The full paper has:

- Additional experiments for an SIR epidemiology model and a Bayesian hierarchical model
- Evaluations with respect to the effective sample size

FULL-LENGTH PAPER AT:
https://arxiv.org/abs/2106.04953!

References

[1] Rainforth, T., Goliński, A., Wood, F., & Zaidi, S. (2020). Target–aware Bayesian inference: how to beat optimal conventional estimators. *Journal of Machine Learning Research*, *21*(88).

[2] https://turing.ml/

Authors


Tim Reichelt


Adam Goliński


Luke Ong


Tom Rainforth

Engineering and Physical Sciences Research Council

EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines & Systems

UNIVERSITY OF OXFORD