OUTPUT-SENSITIVE ADAPTIVE METROPOLIS-HASTINGS FOR PROBABILISTIC PROGRAMS

David Tolpin, Jan Willem van de Meent, Brooks Paige, Frank Wood
University of Oxford
Oxford, UK

PRELIMINARIES

Probabilistic Program

- A program with random computations.
- Distributions are conditioned by `observations'.
- Values of certain expressions are `predicted' — the output.

```
(let [;; Model
      dist (sample (categorical [[normal 1/2] [g
      a (sample (gamma 1 1))
      b (sample (gamma 1 1))
      d (dist a b)]
  ;; Observations
  (observe d 1) (observe d 2)
  (observe d 4) (observe d 7)
  ;; Explanation
  (predict :d (type d))
  (predict :a a) (predict :b b)))
```

Inference Objectives

- Suggest most probable explanation (MPE) - most likely assignment for all non-evidence variable given evidence.
- Approximately compute integral of the form $\Phi = \int_{-\infty}^{\infty} \varphi(x)p(x)dx$
- Continuously and infinitely generate a sequence of samples. ✓

Lighweight Metropolis-Hastings (LMH)

$\mathcal{P}$ — probabilistic program.
$\boldsymbol{x}$ — random variables.
z — output.

Run $\mathcal{P}$ once, remember $\boldsymbol{x}, z$.
loop
  Uniformly select x_i.
  Propose a value for x_i.
  Run $\mathcal{P}$, remember $\boldsymbol{x}', z'$.
  Accept $(\boldsymbol{x}, z = \boldsymbol{x}', z')$
    or reject with MH probability.
  Output z.
end loop

Can we do better?

REFERENCES

1. Christophe Andrieu and Johannes Thoms. A tutorial on adaptive MCMC. Statistics and Computing, 18(4):343–373, 2008.
2. B. Paige, F. Wood, A. Doucet, and Y.W. Teh. Asynchronous anytime sequential Monte Carlo. In NIPS-2014, to appear.
3. David Wingate, Andreas Stuhlmu¨ller, and Noah D. Goodman. Lightweight implementations of probabilistic programming languages via transformational compilation. In Proc. of AISTATS-2011.
4. Frank Wood, Jan Willem van de Meent, and Vikash Mansinghka. A new approach to probabilistic programming inference. In AISTATS-2014.

METROPOLIS HASTINGS WITH ADAPTIVE SCHEDULING

- Selects each x_i with a different probability.
- Maintains vector of weights $\boldsymbol{W}$ of random choices:

Initialize $\boldsymbol{W}^0$ to a constant.
Run $\mathcal{P}$ once.
for $t = 1 \dots \infty$
  Select x_i^t with probability $\alpha_i^t = W_i^t / \sum_{i=1}^{|\boldsymbol{x}^t|} W_i^t$.
  Propose a value for x_i^t.
  Run $\mathcal{P}$, accept or reject with MH probability.
  if accepted
    Compute $\boldsymbol{W}^{t+1}$ based on the *program output*.
  else
    $\boldsymbol{W}^{t+1} \leftarrow \boldsymbol{W}^t$
  end if
end for

QUANTIFYING THE INFLUENCE

- Objective: faster convergence of program output z.
- Adaptation parameter: probabilities of selecting random choices for modification.
- Optimization target: maximize the change in the program output:

$$R^t = \frac{1}{|z^t|} \sum_{k=1}^{|z^t|} \mathbf{1}(z_k^t \neq z_k^{t-1}).$$

W_i reflects the anticipated change in z from modifying x_i.

DELAYED CHANGES

Modifying `x2` affects the output ...

```
(let [x1 (sample (normal 1 10))
      x2 (sample (normal x1 1))]
  (observe (normal x2 1) 2)
  (predict x1))
```

... but only when `x1` is also modified.

BACK-PROPAGATING REWARDS

- For each x_i, reward r_i and count c_i are kept.
- A history of modified random choices is attached to every z_j.

When modification of x_k accepted:

Append x_k to the history.
if $z^{t+1} \neq z^t$
  $w \leftarrow \frac{1}{|history|}$
  for x_m **in** history
    $\bar{r}_m \leftarrow r_m + w,\ c_m \leftarrow c_m + w$
  end for
  Flush the history.
else
  $c_k \leftarrow c_k + 1$
end if

Convergence:

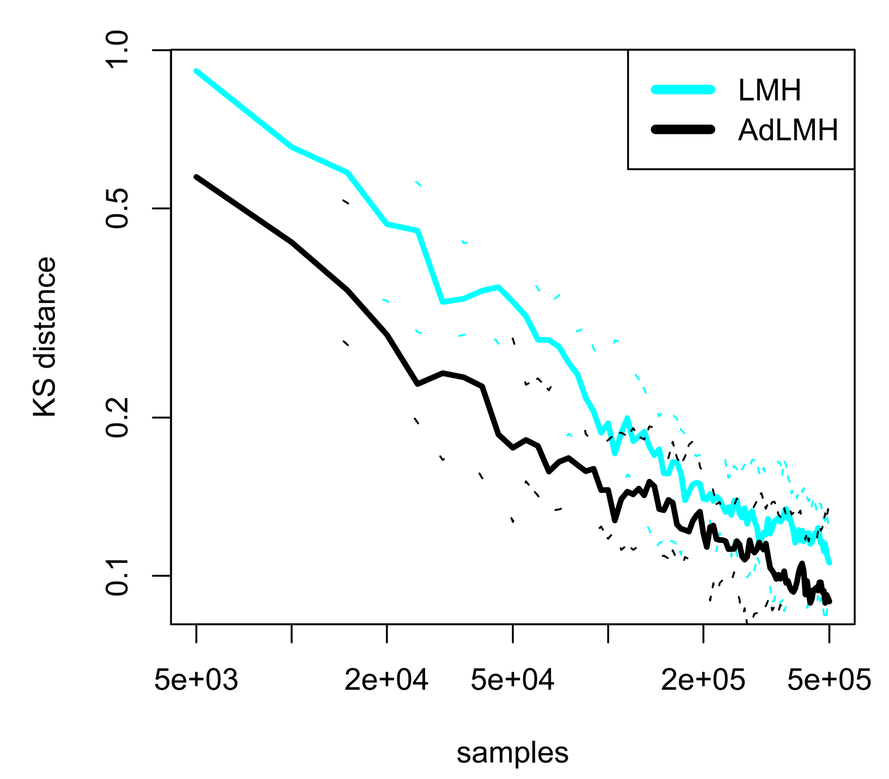
For any partitioning of $\boldsymbol{x}$, Adaptive LMH selects variables from each partition with non-zero probability.

EXPERIMENTS

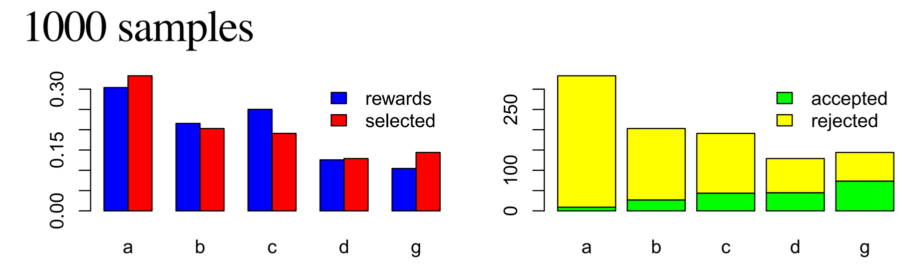
Convergence —Gaussian Process

$f \sim \mathcal{GP}(m, k),$

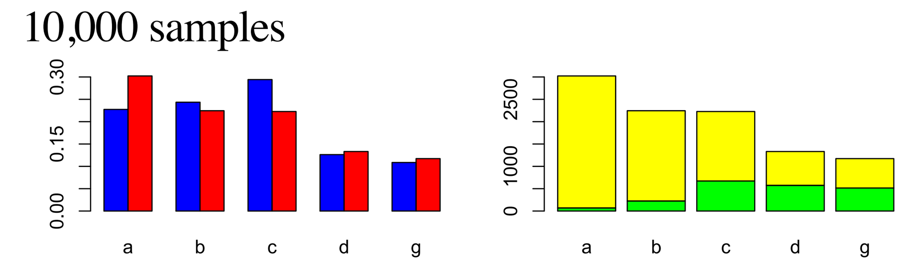
where $m(x) = ax^2 + bx + c, \quad k(x, x') = de^{-\frac{(x-x')^2}{2g}}.$



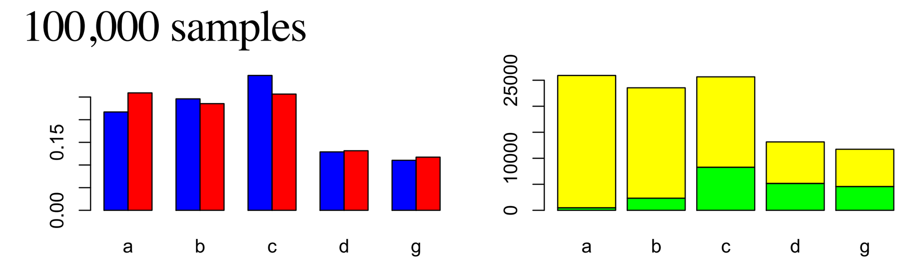
1000 samples



10,000 samples



100,000 samples

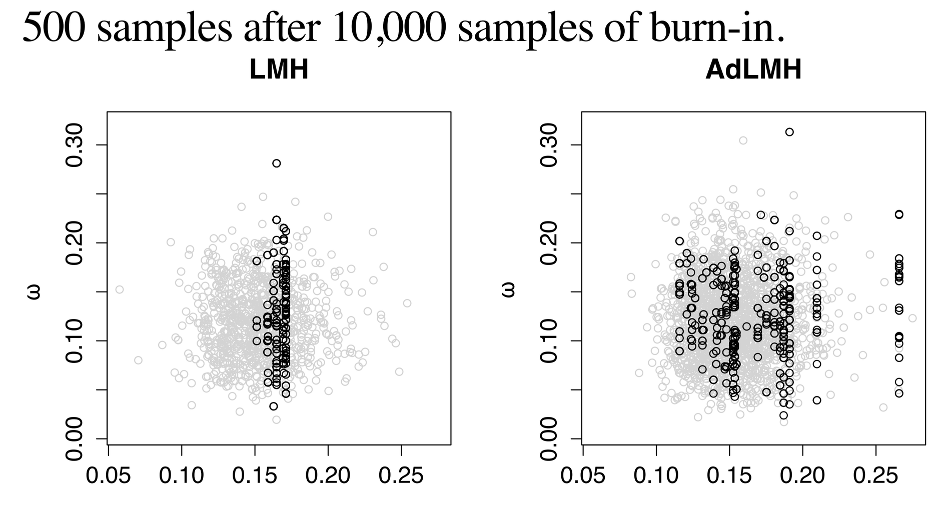


Sample size — Kalman Smoother

$\vec{x}_t \sim \mathrm{Norm}(\vec{A} \cdot \vec{x}_{t-1}, \vec{Q}), \qquad \vec{y}_t \sim \mathrm{Norm}(\vec{C} \cdot \vec{x}_t, \vec{R}).$

$\vec{A} = \begin{bmatrix} \cos\omega & -\sin\omega \\ \sin\omega & \cos\omega \end{bmatrix}, \qquad \vec{Q} = \begin{bmatrix} q & 0 \\ 0 & q \end{bmatrix}.$

100 16-dimensional observations,
500 samples after 10,000 samples of burn-in.



ACKNOWLEDGMENTS

This material is based on research sponsored by DARPA through the U.S. Air Force Research Laboratory under Cooperative Agreement number FA8750-14-2-0004.

CONTRIBUTIONS

- A scheme of rewarding random samples based on program output.
- An approach to propagation of sample rewards to MH proposal scheduling parameters.
- An application of this approach to LMH, where the probabilities of selecting each variable for modification are adjusted.

FUTURE WORK

- Extending the adaptation approach to other sampling methods.
- Reward scheme that takes into account the amount of difference between samples.
- Acquisition of dependencies between predicted expressions and random variables.