Sampling and inference for discrete random probability measures in probabilistic programs

B. Bloem-Reddy* , E. Mathieu*, A. Foster, T. Rainforth, Y. W. Teh, H. Ge, M. Lomelí, Z. Ghahramani

Department of Statistics, University of Oxford & Department of Engineering, University of Cambridge

Overview

- Random probability measures (RPMs) are a cornerstone of Bayesian nonparametric statistics.
- (Prior) sampling from RPMs in probabilistic programming systems (PPSs) is non-trivial since they have a possibly infinite countable support.
- Not all sampling schemes for RPMs are equivalent: we define the *laziest initialization* as the one which avoids unnecessary computation.
- In the special case of the Pitman-Yor process (PYP), a laziest initialization sampling algorithm exists and compares favourably with the *recursive coin-flipping* sampling algorithm. **The runtime of recursive coin-flipping may have infinite expectation**.
- Using a laziest initialization, posterior inference for a Normalized Inverse Gaussian Process (NIGP) mixture model was implemented in the PPS Turing: the first example in a PPS of a Bayesian nonparametric mixture model that is not the Dirichlet process (DP) or the PYP.

Size-biased representation of RPMs

A discrete RPM $\mathbf{P} = (P_j, \Omega_j)_{j\geq 1}$ is a countable collection of *probability weights* and *atoms*. A *size-biased permutation* π of $\mathbf{P}$, denoted $\tilde{\mathbf{P}}$ is a random permutation of the atoms of $\mathbf{P}$ such that [1]

$$(\widetilde{P}_1, \widetilde{\Omega}_1) = (P_{\pi(1)}, \Omega_{\pi(1)}) \quad \text{where} \quad \mathbb{P}(\pi(1) = j \mid P_1, P_2, \dots) = P_j$$

$$(\widetilde{P}_2, \widetilde{\Omega}_2) = (P_{\pi(2)}, \Omega_{\pi(2)}) \quad \text{where} \quad \mathbb{P}(\pi(2) = j \mid \pi(1), P_1, P_2, \dots) = \frac{P_j}{1 - \widetilde{P}_1} ,$$

and so on.

We define a *laziest initialization* as one instantiating the minimal number of atoms used by the sample. $\widehat{\mathbf{P}}$ is a laziest initialization of $\mathbf{P}$ while sampling $X_1, X_2, \dots$ iff it is a size-biased representation of $\mathbf{P}$ induced by $X_1, X_2, \dots$

The stick-breaking construction of the PYP is distributionally equivalent to the size-biased representation of the PYP [2, 1]: $\widetilde{P}_j \overset{\text{d}}{=} V_j \prod_{i=1}^{j-1}(1 - V_i)$ for each j. Hence the predictive distribution of X_{n+1} given $\widetilde{\mathbf{P}}_{K_n}$ is

$$\mathbb{P}[X_{n+1} \in \bullet \mid \widetilde{\mathbf{P}}_{K_n}] = \textstyle\sum_{j=1}^{K_n} \widetilde{P}_j \delta_{\widetilde{\Omega}_j}(\bullet) + \left(1 - \sum_{j=1}^{K_n} \widetilde{P}_j\right) H_0(\bullet) \ . \qquad (1)$$

Two generative sampling algorithms for RPMs

$\mathbf{X}_n$ can therefore be sampled from a PYP prior with parameters $\alpha \in (0, 1)$, $\theta > -\alpha$ and base measure H_0 as follows:

Algorithm 1 Recursive coin-flipping for sampling from the PYP

$M = 0$ ▷ For tracking the number of atoms initialized.
for $i = 1 \colon n$ **do** ▷ Iterate over observations.
  $j = 0$, coin $= 0$
  while coin==0 **do** ▷ Recursively (in j) flip V_j-coins until the first heads.
    $j = j + 1$
    if $j > M$ **then** ▷ Instantiate V_j and Ω_j when necessary.
      $V_j \sim \text{Beta}(1 - \alpha, \theta + j\alpha)$, $\Omega_j \sim H_0$
      $M = M + 1$
    end if
    coin $\sim \text{Bernoulli}(V_j)$ ▷ Flip a V_j-coin.
  end while
  $X_i = \Omega_j$ ▷ X_i takes the value of the atom corresponding to the first heads.
end for

Algorithm 2 Laziest initialization for sampling from the PYP

$K = 0$ ▷ For tracking the number of atoms initialized.
for $i = 1 \colon n$ **do** ▷ Iterate over observations.
  new $=$ true ▷ Should a new atom be created on this iteration?
  for $k = 1 \colon K$ **do** ▷ Iterate over existing atoms
    coin $\sim \text{Bernoulli}(V_j)$ ▷ Flip a V_j-coin.
    if coin $== 1$ **then**
      $X_i = \Omega_k$ ▷ X_i takes the value atom k
      new $=$ false ▷ A new atom is not required
      break
    end if
  end for
  if new **then** ▷ None of the K existing atoms was selected
    $K = K + 1$
    $V_K \sim \text{Beta}(1 - \alpha, \theta + K\alpha)$, $\Omega_K \sim H_0$ ▷ Initialize a new atom
    $X_i = \Omega_K$ ▷ X_i takes the value of the new atom
  end if
end for

For more general RPMs the algorithms are similar, though one typically has to consider the *total mass* T of the unnormalized measure.

Number of atoms instantiated for the PYP

Continuing with the PYP, let M_n and K_n be the number of atoms instantiated by recursive coin-flipping and laziest instantiation respectively. The first proposition shows that M_n may have infinite expectation.

***Proposition* 1.** *Let M_n be the number of atoms instantiated by the recursive coin-flipping scheme to sample $\mathbf{X}_n$. Then $\mathbb{E}_{\alpha,\theta}[M_1] < \infty$ if and only if $\alpha < \frac{1}{2}$. Furthermore, for all $n \geq 1$, if M_n is finite then $\mathbb{E}_{\alpha,\theta}[M_{n+1} \mid M_n] < \infty$ if and only if $\alpha < \frac{1}{2}$.*

By contrast, $K_n \leq n$ for all α, θ.

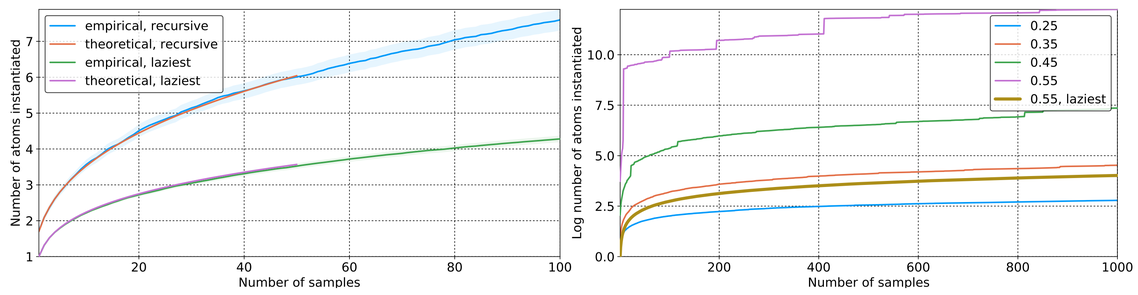

Figure 1: The expected number of atoms generated by recursive coin-flipping and by induced size-biased (laziest) schemes when sampling $\mathbf{X}_n$ from a PYP. Empirical means and standard errors were generated via simulation because theoretical values are numerically unstable for $n > 50$. Left: $\alpha = 0.25$, $\theta = 0.1$. Right: Empirical means (for 4,000 simulations) for $\theta = 0.1$ and various α for the recursive coin-flipping scheme. (Note the log scale on the vertical axis.)

Inference for RPM mixture models in PPSs

We consider RPM mixture models for observations $(Y_i)_{1\leq i\leq n}$

$$\mathbf{P} \sim \mu$$
$$X_i \mid \mathbf{P} \sim \mathbf{P}$$
$$Y_i \mid X_i \sim \mathcal{F}(\cdot \mid X_i)$$

where $\mathcal{F}$ is a known emission distribution parametrised by X_i.

In our experiments, μ was either a NIGP or a PYP, and sampling was implemented using a laziest initialization algorithm based on their size-biased representations. We ran our experiments on a Gaussian mixture model with shared variance, using the Galaxy dataset. The generative model was implemented in Turing [3] and inference was performed using its Sequential Monte Carlo (SMC) [4] model agnostic inference algorithm.

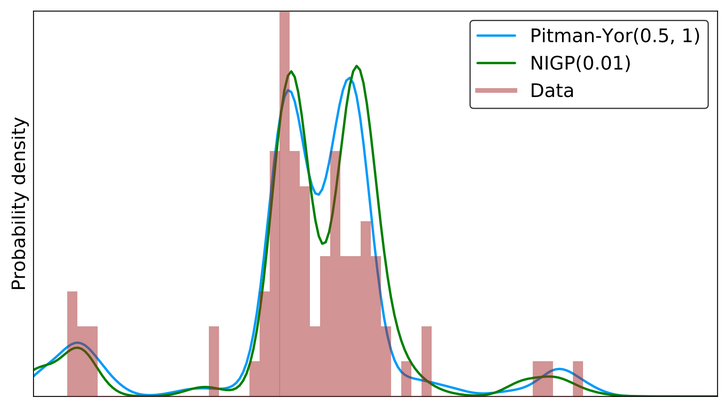

Figure 2: Visualizations of the estimated posterior predictive distribution of the PYP and NIGP mixture models fit to the galaxy data set.

Future work

- Gibbs sampler: split updates for atoms location, sticks length, assignments and hyperparameters.
- Variational inference using laziest initialization.

Acknowledgements

- EPSRC, ERC, MSR & Google.

References

[1] Jim Pitman. *Combinatorial Stochastic Processes*, volume 1875 of *Ecole d'Eté de Probabilités de Saint-Flour*. Springer-Verlag Berlin Heidelberg, 2006.

[2] Jim Pitman. Random discrete distributions invariant under size-biased permutation. *Advances in Applied Probability*, 28(2):525–539, 1996.

[3] Hong Ge, Kai Xu, Adam Scibior, Zoubin Ghahramani, et al. The Turing language for probabilistic programming. June 2016.

[4] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.