Density function estimation using ergodic recursion

Erik Bodin¹, Zhenwen Dai², Neill D. F. Campbell³, Carl Henrik Ek⁴

¹University of Bristol  ²Spotify Research  ³University of Bath  ⁴University of Cambridge

Summary

- Physically motivated likelihood functions and priors often give rise to complicated or degenerate inference problems, often requiring weeks of computation and a portfolio of methods even in relatively low dimension.
- We propose a fast, robust and simple-to-use algorithm, for general Bayesian computation and inference, applicable to a wide set of problems.
  - Robust to degeneracies in the density function, such as discontinuities, nonsmoothness, and zero density regions.
  - Does not rely on gradients.
  - Asymptotically exact.
- Enables down-stream tasks, such as
  - Bayesian parameter inference,
  - evidence estimation,
  - expectations of functions,
  - fast, constant-time sampling,
  - conditional and marginal distribution estimation,
  - estimation of divergences and mutual information.

Methodology

\[ P(\theta) = \frac{f(\theta)}{\int f(\theta) \, d\theta} = \frac{f(\theta)}{Z}, \]

where \( f : \Omega \to \mathbb{R}_+ \) is an explicitly unknown density function defined on a bounded sample space \( \Omega \subset \mathbb{R}^D \), which can be evaluated for any \( \theta \in \Omega \). \( Z \) is the unknown normalizing constant.

Representation of \( \hat{f} \)

- Search-tree over partitions of the sample space,
- constituting a Riemann sum.

Iterative construction of \( \hat{f} \)

- Addresses sample-efficiency by the prioritization-order of refining partitions.
- Uses an exploration-exploitation trade-off which considers all density observations seen so far.
- The time complexity of acquiring a new partition at step \( t \) is \( \mathcal{O}(\log N_t + U_t \log U_t) \).

In practice, decision-times are fractions of a millisecond even after millions of partitions.

Input: Density function \( f \) defined over \( \Omega \) with unknown normalization constant \( Z \)
Output: Approximation \( \hat{f}, \hat{Z} \), as specified by the produced partitioning \( \Pi_t \)

1. set \( t = 1 \) and initial partitioning \( \Pi_1 = \{\Omega\} \);
2. while \( N_t \leq N_{\text{max}} \) do
4. divide each partition \( \Pi_t \), each one resulting in \( \{\Omega_i\} \) new partitions;
5. add all sets of \( \{\Omega_i\} \) into \( \Pi_t \), remove the divided partitions \( \{\Omega_i\} \) from \( \Pi_t \);
6. set \( t \to t + 1 \) and update data structures;
7. end
8. return \( \Pi_t \);

Partition division criteria

\[ V_i \cdot f(\theta_i) + K_d \frac{d_i}{2} \geq V_i \cdot f(\theta_i) + K_d \frac{d_i}{2}, \quad \forall i \in [1, N_t] \]

and \( V_i \cdot f(\theta_i) + K_d \frac{d_i}{2} \geq \frac{Z}{N_t + 1} \)

\[ K^{\text{new}} = \frac{V_{i+1}(f(\theta_{i+1}) - V_{i+1}(f(\theta_i)))}{V_{i+1} - V_i} \]

Contact Information

www.linkedin.com/in/stigerikbodin
mail@erikbodin.com