ARSpy Documentation

Release 0.3

Moritz Freidank

Nov 01, 2019

Contents:

CHAPTER 1

ARSPY

This package provides a pure python/numpy implementation of adaptive rejection sampling as proposed by P. Wild, W.R. Gilks in *Algorithm AS 287: Adaptive Rejection Sampling from Log Concave Density functions*.

Under the (frequently satisfied) assumption that the target distribution to sample from has a log-concave density function, this algorithm allows us to sample without calculating any integrals.

This sampling method is *exact* (all resulting samples are i.i.d) and *efficient* and our implementation can handle any univariate log-concave distribution.

One prime use case is Gibbs sampling, where one frequently encounters many 1D log-concave distributions.

CHAPTER 2

Install

Simply run:

```
pip3 install ARSpy
```

2.1 API Reference

2.1.1 Adaptive Rejection Sampling

This module contains a python/numpy implementation of Adaptive Rejection Sampling (ARS) introduced by Gilks and Wild in 1992.

Our implementation considers both lower and upper hull as introduced in the original paper. This allows us to extract larger amounts of samples at a time in a stable way. Furthermore, we do *not* require the *derivative* of the logpdf as input to our sampler, only the logpdf itself.

Our code is a port of an original matlab code in pmtk3 by Daniel Eaton (danieljameseaton@gmail.com) and compared to an open-source julia port (by Levi Boyles) of the same matlab function for testing purposes.

`arspy.ars.`**`adaptive_rejection_sampling`**(*logpdf: <built-in function callable>, a: float, b: float, domain: Tuple[float, float], n_samples: int, random_stream=None*)

Adaptive rejection sampling samples exactly (all samples are i.i.d) and efficiently from any univariate log-concave distribution. The basic idea is to successively determine an envelope of straight-line segments to construct an increasingly accurate approximation of the logarithm. It does not require any normalization of the target distribution.

Parameters

logpdf: callable Univariate function that computes $log(f(u))$ for a given u, where $f(u)$ is proportional to the target density to sample from.

a: float Lower starting point used to initialize the hulls. Must lie in the domain of the logpdf and it must hold: $a < b$.

b: float Upper starting point used to initialize the hulls. Must lie in the domain of the logpdf and it must hold: $a < b$.

domain [Tuple[float, float]] Domain of *logpdf*. May be unbounded on either or both sides, in which case *(float("-inf"), float("inf"))* would be passed. If this domain is unbounded to the left, the derivative of the logpdf for x<= a must be positive. If this domain is unbounded to the right the derivative of the logpdf for x>=b must be negative.

n_samples: int Number of samples to draw.

random_stream [RandomState, optional] Seeded random number generator object with same interface as a NumPy RandomState object. Defaults to *None* in which case a NumPy RandomState seeded from */dev/urandom* if available or the clock if not will be used.

Returns

samples [list] A list of samples drawn from the target distribution f with the given *logpdf*.

Examples

Sampling from a simple gaussian, adaptive rejection sampling style. We use the logpdf of a standard gaussian and this small code snippet demonstrates that our sample approximation accurately approximates the mean:

```
>>> from math import isclose
>>> from numpy import log, exp, mean
>>> gaussian_logpdf = lambda x, sigma=1: log(exp(-x ** 2 / sigma))
>>> a, b = -2, 2  # a < b must hold
>>> domain = (float("-inf"), float("inf"))
>>> n_samples = 10000
>>> samples = adaptive_rejection_sampling(logpdf=gaussian_logpdf, a=a, b=b,␣
↪domain=domain, n_samples=n_samples)
>>> isclose(mean(samples), 0.0, abs_tol=1e-02)
True
```

2.1.2 Hull

`arspy.hull.`**`compute_hulls`**(*S*, *fS*, *domain*)

(Re-)compute upper and lower hull given the segment points *S* with function values *fS* and the *domain* of the logpdf.

Parameters

S [np.ndarray (N, 1)] Straight-line segment points accumulated thus far.

fS [tuple] Value of the *logpdf* under sampling for each of the given segment points in *S*.

domain [Tuple[float, float]] Domain of *logpdf*. May be unbounded on either or both sides, in which case *(float("-inf"), float("inf"))* would be passed. If this domain is unbounded to the left, the derivative of the logpdf for x<= a must be positive. If this domain is unbounded to the right the derivative of the logpdf for x>=b must be negative.

Returns

lower_hull: List[arspy.hull.HullNode]

upper_hull: List[arspy.hull.HullNode]

`arspy.hull.`**`sample_upper_hull`**(*upper_hull*, *random_stream*)

Return a single value randomly sampled from the given *upper_hull*.

Parameters

upper_hull [List[pyars.hull.HullNode]] Upper hull to evaluate.

random_stream [numpy.random.RandomState] (Seeded) stream of random values to use during sampling.

Returns

sample [float] Single value randomly sampled from *upper_hull*.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index