a minimax near-optimal algorithm for adaptive rejection sampling

Juliette Achddou, Joseph Lam-Weil, Alexandra Carpentier, Gilles Blanchard

March 2019

Numberly Group, Otto Von Guericke University

Motivation

Context

- Goal: sample independently from a distribution which admits a density f.
- Evaluations of f are possible but costly.
- Simple rejection sampling would incur a large number of unused evaluations.

Assumption f is positively lower bounded, with bounded (known) support, and (s, H)-Hölder $(0 < s \leq 1)$:

$$\forall x, y \in [0,1]^d, \; |f(x) - f(y)| \leq H\|x - y\|_\infty^s$$

Rejection Sampling

Definitions Let

- f be the density you **wish to sample from**. (target density)
- g be a density that is **easy to sample from**. (proposal density)
- M be a constant such that $Mg \geq f$. (rejection constant)

Rejection Sampling

Algorithm 1: Rejection Sampling Algorithm

input : M, g s.t. $f \leq Mg$
output: samples S
for $t = 1,\dots,n$ **do**
  Sample Y according to g and U according to $\mathcal{U}_{[0,1]}$;
  If $U \leq \frac{f(Y)}{Mg(Y)}$, add Y to S ;
end

$\mathcal{S}$ contains independent samples drawn from f.
The acceptance rate is $\frac{1}{M}$.

Rejection Sampling

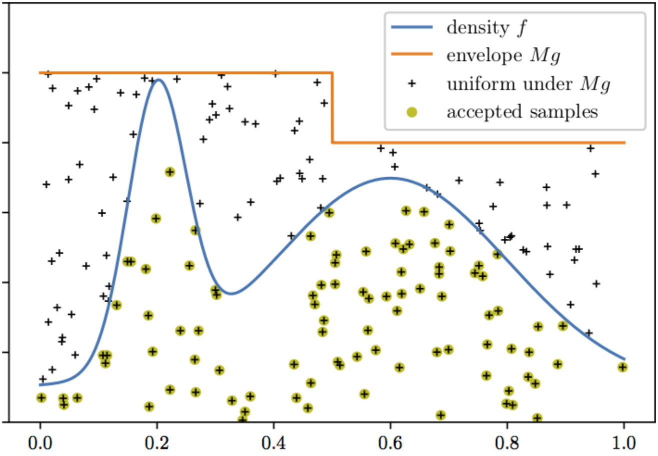


Figure: Illustration of Rejection Sampling

Adaptive Rejection Sampling

Let n be the budget. Let $\mathcal{S} = \emptyset$ be the set of samples from f.

Procedure At each step $t \leq n$,

Adaptive Rejection Sampling

Let n be the budget. Let $\mathcal{S} = \emptyset$ be the set of samples from f.

Procedure At each step $t \leq n$,

1. $\{(X_1, f(X_1)), \ldots, (X_{t-1}, f(X_{t-1}))\}$ are known.

Adaptive Rejection Sampling

Let n be the budget. Let $\mathcal{S} = \emptyset$ be the set of samples from f.

Procedure At each step $t \leq n$,

1. $\{(X_1, f(X_1)), \ldots, (X_{t-1}, f(X_{t-1}))\}$ are known.
2. g_t, M_t are chosen.

Adaptive Rejection Sampling

Let n be the budget. Let $\mathcal{S} = \emptyset$ be the set of samples from f.

Procedure At each step $t \leq n$,

1. $\{(X_1, f(X_1)), \ldots, (X_{t-1}, f(X_{t-1}))\}$ are known.
2. g_t, M_t are chosen.
3. A rejection sampling step is done using (g_t, M_t). This generates X_t.

Adaptive Rejection Sampling

Let n be the budget. Let $\mathcal{S} = \emptyset$ be the set of samples from f.

Procedure At each step $t \leq n$,

1. $\{(X_1, f(X_1)), \ldots, (X_{t-1}, f(X_{t-1}))\}$ are known.
2. g_t, M_t are chosen.
3. A rejection sampling step is done using (g_t, M_t). This generates X_t.
4. If X_t is not rejected, it is added to $\mathcal{S}$.

Adaptive Rejection Sampling

Let n be the budget. Let $\mathcal{S} = \emptyset$ be the set of samples from f.

Procedure At each step $t \leq n$,

1. $\{(X_1, f(X_1)), \ldots, (X_{t-1}, f(X_{t-1}))\}$ are known.
2. g_t, M_t are chosen.
3. A rejection sampling step is done using (g_t, M_t). This generates X_t.
4. If X_t is not rejected, it is added to $\mathcal{S}$.

Definition of the loss $L_n = n - \#\mathcal{S} \times \mathbf{1}\{\forall t \leq n : \ f \leq M_t g_t\}$.

Our contributions

1. Nearest Neighbor Adaptive Rejection Sampling (NNARS).

Our contributions

1. Nearest Neighbor Adaptive Rejection Sampling (NNARS).
2. Minimax lower bound.

Our contributions

1. Nearest Neighbor Adaptive Rejection Sampling (NNARS).
2. Minimax lower bound.
3. NNARS is minimax near-optimal.

Minimax optimality

Let

- $\mathcal{A}$ be the set of ARS algorithms.
- $\mathcal{F}_0$ be the set of densities: positively lower bounded, with bounded support, and (s, H)-Hölder ($0 < s \leq 1$):

$$\forall x, y \in [0,1]^d,\ |f(x) - f(y)| \leq H\|x - y\|_\infty^s$$

Minimax rate

$$\inf_{A \in \mathcal{A}} \sup_{f \in \mathcal{F}_0} \frac{L_n(A; f)}{n}.$$

Minimax optimality

Let

- $\mathcal{A}$ be the set of ARS algorithms.
- $\mathcal{F}_0$ be the set of densities: positively lower bounded, with bounded support, and (s, H)-Hölder $(0 < s \leq 1)$:

$$\forall x, y \in [0,1]^d,\ |f(x) - f(y)| \leq H\|x - y\|_\infty^s$$

Minimax rate

$$\varphi_n^* = \inf_{A \in \mathcal{A}} \sup_{f \in \mathcal{F}_0} \frac{L_n(A; f)}{n}.$$

Nearest Neighbor Adaptive Rejection Sampling.
Divide the n steps into K rounds, where each round k contains double the amount of steps than $k-1$.

Obtaining an upper bound/Focus on one algorithm

Nearest Neighbor Adaptive Rejection Sampling.
Divide the n steps into K rounds, where each round k contains double the amount of steps than $k-1$.

At each round $0 \leq k \leq K-1$:

- Use an estimator $\hat{f}_k$ of f based on the previous evaluations.
- Take $M_{(k+1)} g_{(k+1)} = \hat{f}_k + \hat{r}_k$, where $\hat{r}_k$ is a confidence bound for $|\hat{f}_k - f|$.

Approximate nearest neighbor estimator $\hat{f}_k$

At round k,

- we know $\{(X_1, f(X_1)), \ldots, (X_{N_k}, f(X_{N_k}))\}$.
- build a uniform grid of $\sim N_k$ cells with side-length $\sim {N_k}^{-1/d}$.

Let us determine $\hat{f}_k(x)$.

Approximate nearest neighbor estimator $\hat{f}_k$

At round k,

- we know $\{(X_1, f(X_1)), \ldots, (X_{N_k}, f(X_{N_k}))\}$.
- build a uniform grid of $\sim N_k$ cells with side-length $\sim {N_k}^{-1/d}$.

Let us determine $\hat{f}_k(x)$.

1. x is in the l-th cell.

Approximate nearest neighbor estimator $\hat{f}_k$

At round k,

- we know $\{(X_1, f(X_1)), \ldots, (X_{N_k}, f(X_{N_k}))\}$.
- build a uniform grid of $\sim N_k$ cells with side-length $\sim {N_k}^{-1/d}$.

Let us determine $\hat{f}_k(x)$.

1. x is in the l-th cell.
2. Let X_i be the nearest neighbor of the center of the l-th cell.

Approximate nearest neighbor estimator $\hat{f}_k$

At round k,

- we know $\{(X_1, f(X_1)), \ldots, (X_{N_k}, f(X_{N_k}))\}$.
- build a uniform grid of $\sim N_k$ cells with side-length $\sim {N_k}^{-1/d}$.

Let us determine $\hat{f}_k(x)$.

1. x is in the l-th cell.
2. Let X_i be the nearest neighbor of the center of the l-th cell.
3. Then $\hat{f}_k(x) = f(X_i)$.

Approximate nearest neighbor estimator $\hat{f}_k$

Why approximate?

- Direct 1-NN $\Rightarrow$ Voronoi cells.

Approximate nearest neighbor estimator $\hat{f}_k$

Why approximate?

- Direct 1-NN $\Rightarrow$ Voronoi cells.
- Approximate NN $\Rightarrow$ hypercubes.

Approximate nearest neighbor estimator $\hat{f}_k$

Why approximate?

- Direct 1-NN $\Rightarrow$ Voronoi cells.
- Approximate NN $\Rightarrow$ hypercubes.

Then

$$g_{k+1} : x \to \frac{\hat{f}_k(x) + \hat{r}_k}{M_{k+1}} \quad \text{is } \textbf{easy to sample from}.$$

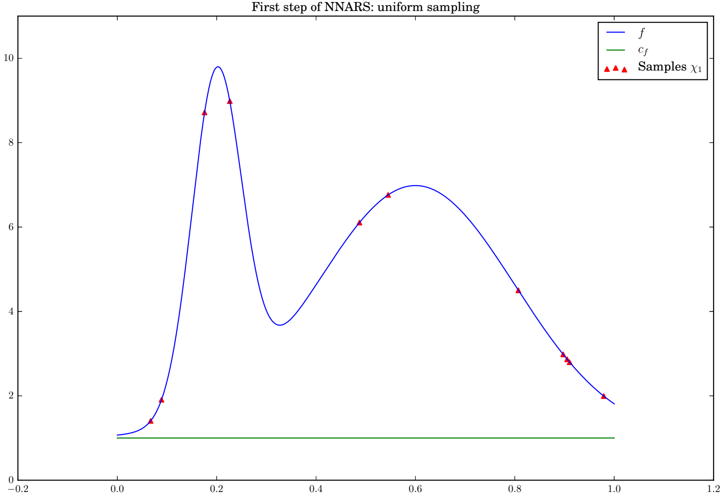
Approximate nearest neighbor estimator $\hat{f}_k$

Confidence term

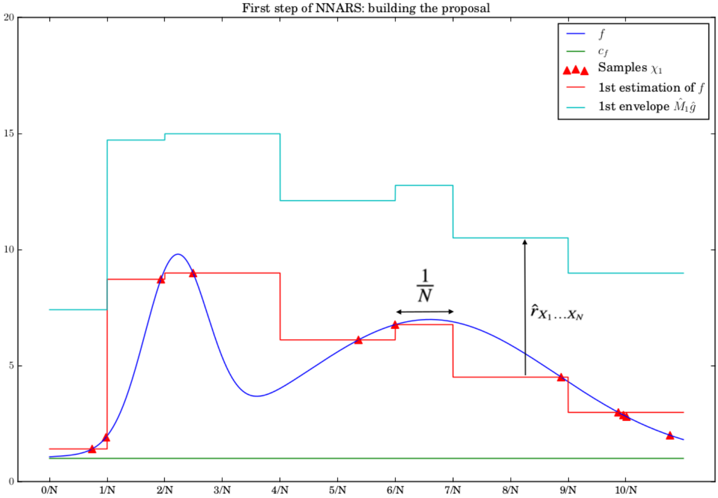
$$\hat{r}_k = H\left(\max_{u \in \mathcal{C}_{N_k}} \min_{i \leq N_k} \|u - X_i\|_\infty + \frac{1}{2(\lfloor N_k{}^{\frac{1}{d}} \rfloor + 1)}\right)^s.$$

$$g_{k+1} : x \to \frac{\hat{f}_k(x) + \hat{r}_k}{M_{k+1}}, \text{ where } M_{k+1} \text{is } \int_0^1 f_k(x)dx + \hat{r}_k$$

The algorithm: NNARS



The algorithm: NNARS



The bounds obtained

Assume n is large enough.

Upper bound

$$
\begin{aligned}
\mathbb{E}_f L_n(\text{NNARS}) &\leq \frac{20}{2^{1-s/d}-1} c_f{}^{-2}(1+\sqrt{2\log 3n}) \log^{s/d}(5n) n^{1-s/d} \\
&+ \quad (25+40+2(10Hc_f^{-1})^{d/s}) c_f^{-1} \log^2(n) \\
&= O(\log^2(n) n^{1-s/d}),
\end{aligned}
$$

The bounds obtained

Assume n is large enough.

Upper bound

$$\begin{aligned}\mathbb{E}_f L_n(\text{NNARS}) &\leq \frac{20}{2^{1-s/d}-1} c_f^{-2}(1+\sqrt{2\log 3n})\log^{s/d}(5n)n^{1-s/d} \\ &+ \quad (25+40+2(10Hc_f^{-1})^{d/s})c_f^{-1}\log^2(n) \\ &= O(\log^2(n)n^{1-s/d}),\end{aligned}$$

Lower bound

$$\begin{aligned}\inf_{A\in\mathcal{A}} \sup_{f\in\mathcal{F}_0\cap\{f:I_f=1\}} \mathbb{E}_f(L_n(A)) &\geq 3^{-1}2^{-1-3s-2d}5^{-s/d}n^{1-s/d} \\ &= O(n^{1-s/d}).\end{aligned}$$

Experiment

Setup

- A forest fire data set with 13 attributes and 517 observations. [Cortez and Morais(2007)]
- Focus on 2 attributes: Duff Moisture Code (DMC) and Drought Code (DC).

Goal: Generate more data from the underlying distribution of the bivariate random variable (DMC,DC).

Experiment

Preprocessing

- Rescale data to $[0, 1]^2$.
- Created a density using the Epanechnikov kernel:

$$K(u) = \frac{3}{4}(1 - u^2)$$

for any $|u| \leq 1$.

Experiment

Preprocessing

- Rescale data to $[0,1]^2$.
- Created a density using the Epanechnikov kernel:

$$K(u) = \frac{3}{4}(1 - u^2)$$

for any $|u| \leq 1$.

Results

n=10^5, 2D	sampling rate
NNARS	$45.7\% \pm 0.1\%$
Pliable RS	$16.0\% \pm 0.1\%$
Simple RS	$15.5\% \pm 0.1\%$

Table: Sampling rates for forest fires data [Cortez and Morais(2007)]

Summary of the contributions

- A **minimax lower bound** was found for the **adaptive rejection sampling** problem.
- NNARS is a **near-optimal** adaptive rejection sampling algorithm.
- NNARS does well **experimentally**.

Ressources

J. Achdou, J. C. Lam, A. Carpentier, and G. Blanchard.
A minimax near-optimal algorithm for adaptive rejection sampling.
ArXiv e-prints, October 2018.

P. Cortez and A. J. R. Morais.
A data mining approach to predict forest fires using meteorological data.
2007.

Github: jlamweil/NNARS

Questions?