A MINIMAX NEAR-OPTIMAL ALGORITHM FOR ADAPTIVE REJECTION SAMPLING

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

March 2019

Otto-von-Guericke Magdeburg University

Rejection Sampling

Definitions Let

- f be the density you **wish to sample from**. (target density)

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)

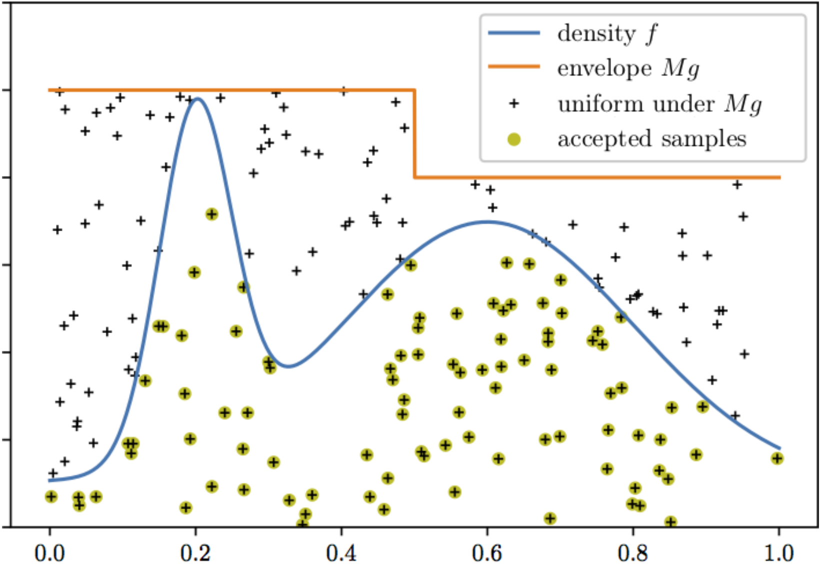


Figure: Illustration of Rejection Sampling

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 $t \leq n$,

- $\{(X_1, f(X_1)), \ldots, (X_{t-1}, f(X_{t-1}))\}$ 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 $t \leq n$,

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

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

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

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

$$\frac{L_n(A; f)}{n}.$$

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

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

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

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

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 twice the number of steps round $k-1$ has.

Nearest Neighbor Adaptive Rejection Sampling.
Divide the n steps into K rounds, where each round k contains twice the number of steps round $k-1$ has.

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 $\hat{f}_k(x) = f(X_i)$.

Choice of parameter

Nearest Neighbor Estimator:

- Choice of an **optimal number of neighbors**?

Choice of parameter

Nearest Neighbor Estimator:

- Choice of an **optimal number of neighbors**?
- **Noiseless** setting $\Rightarrow$ 1-NN is optimal.

Choice of parameter

Nearest Neighbor Estimator:

- Choice of an **optimal number of neighbors**?
- **Noiseless** setting $\Rightarrow$ 1-NN is optimal.

Optimal bandwidth for a Kernel Estimator:

- **Noisy** setting: $h = N^{-1/(d+2s)}$.
- **Noiseless** setting: $h = N^{-1/d}$.

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.

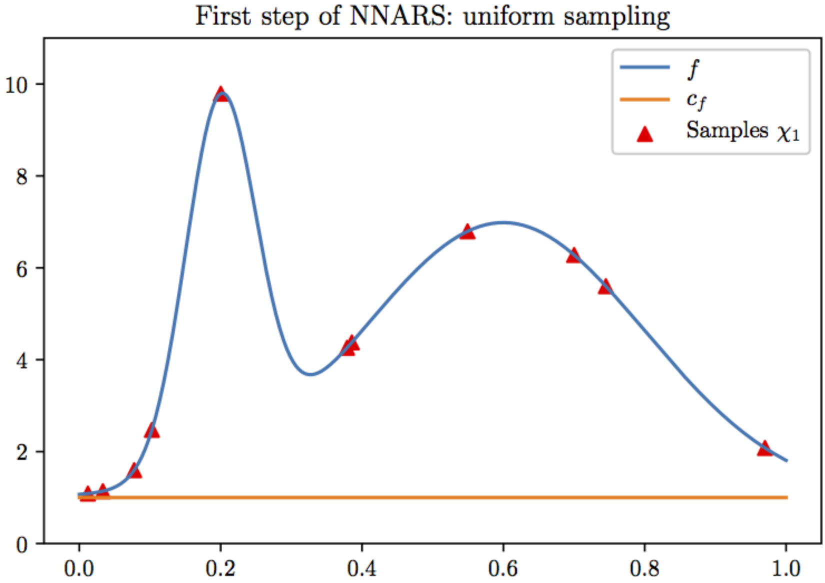
APPROXIMATE NEAREST NEIGHBOR ESTIMATOR $\hat{f}_k$

Why approximate?

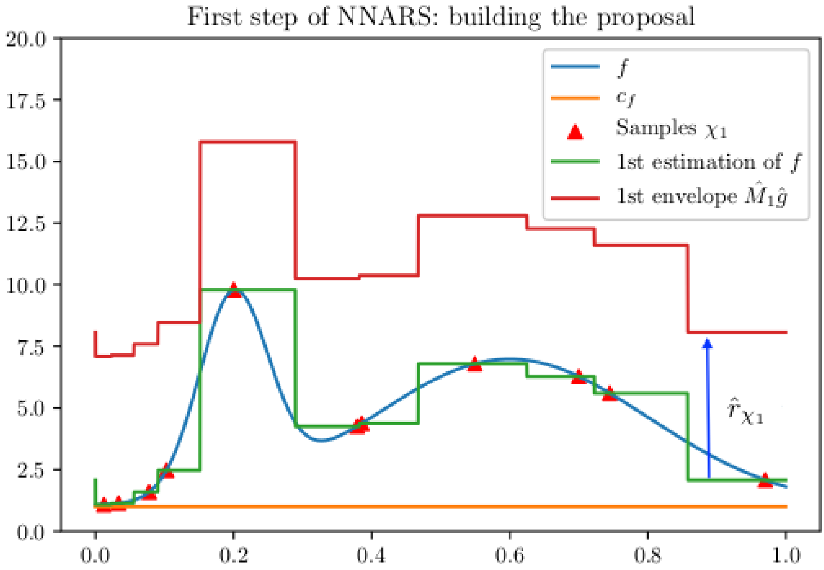
- Direct 1-NN $\Rightarrow$ Voronoi cells.
- Approximate NN $\Rightarrow$ hypercubes.

Then

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


First step of NNARS: uniform sampling
f
c_f
Samples χ_1
10
8
6
4
2
0
0.0
0.2
0.4
0.6
0.8
1.0

The algorithm: NNARS



The bounds obtained

Assume n is large enough.

Upper bound

$$\begin{aligned}\mathbb{E}_f L_n(\text{NNARS}) &\leq 40Hc_f^{-1}(1+\sqrt{2\log(3n)})(\log(2n))^{s/d}n^{1-s/d} \\ &\quad + \left(25+80c_f^{-1}+2(10H)^{d/s}c_f^{-1-d/s}\right)\log^2(n) \\ &= O(\log^2(n)n^{1-s/d}).\end{aligned}$$

Assume n is large enough.

Upper bound

$$\begin{aligned}\mathbb{E}_f L_n(\text{NNARS}) &\leq 40Hc_f^{-1}(1+\sqrt{2\log(3n)})(\log(2n))^{s/d}n^{1-s/d} \\ &\quad + \left(25+80c_f^{-1}+2(10H)^{d/s}c_f^{-1-d/s}\right)\log^2(n) \\ &= O(\log^2(n)n^{1-s/d}).\end{aligned}$$

Lower bound

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

Obtaining the lower bound

Simpler setting. An algorithm in $\mathcal{A}'$ chooses

1. n **points** in order to evaluate them with f.
2. **an envelope** in order to sample n other points using rejection sampling.

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

Simpler setting. An algorithm in $\mathcal{A}'$ chooses

1. n **points** in order to evaluate them with f.
2. **an envelope** in order to sample n other points using rejection sampling.

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

Obtaining the lower bound

Reduction of the space

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

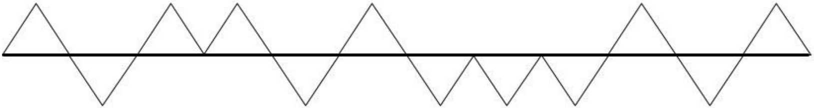
Obtaining the lower bound

Reduction of the space

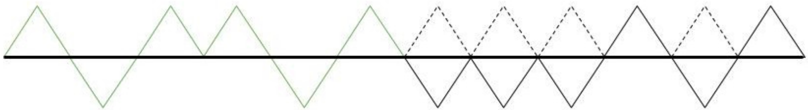
$$\inf_{A \in \mathcal{A}'} \sup_{f \in \mathcal{F}_0} \frac{L_n(A; f)}{n} \geq \inf_{A \in \mathcal{A}'} \frac{\mathbb{E}_{f \sim D_{\mathcal{F}_0}} L_n(A; f)}{n}.$$

Reduction of the space

$$\inf_{A \in \mathcal{A}'} \sup_{f \in \mathcal{F}_0} \frac{L_n(A; f)}{n} \geq \inf_{A \in \mathcal{A}'} \frac{\mathbb{E}_{f \sim D_{\mathcal{F}_0}} L_n(A; f)}{n}.$$

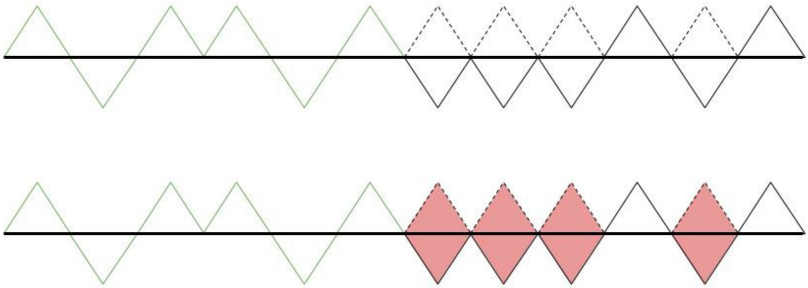


Reduction of the space



Obtaining the lower bound

Reduction of the space



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**.

Resources

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

Github: jlamweil/NNARS

Questions?