Pliable Rejection Sampling

Akram Erraqabi[1] AKRAM.ER-RAQABI@UMONTREAL.CA
MILA, Université de Montréal, Montréal, QC H3C 3J7, Canada

Michal Valko MICHAL.VALKO@INRIA.FR
INRIA Lille - Nord Europe, SequeL team, 40 avenue Halley 59650, Villeneuve d'Ascq, France

Alexandra Carpentier CARPENTIER@MATH.UNI-POTSDAM.DE
Institut für Mathematik, Univertät Potsdam, Germany, Haus 9 Karl-Liebknecht-Strasse 24-25, D-14476 Potsdam

Odalric-Ambrym Maillard ODALRIC.MAILLARD@INRIA.FR
INRIA Saclay - Île-de-France, TAO team, 660 Claude Shannon, Université Paris Sud, 91405 Orsay, France

Abstract

Rejection sampling is a technique for sampling from difficult distributions. However, its use is limited due to a high rejection rate. Common adaptive rejection sampling methods either work only for very specific distributions or without performance guarantees. In this paper, we present *pliable rejection sampling* (PRS), a new approach to rejection sampling, where we learn the sampling proposal using a kernel estimator. Since our method builds on rejection sampling, the samples obtained are with high probability i.i.d. and distributed according to f. Moreover, PRS comes with a guarantee on the number of accepted samples.

1. Introduction

In machine learning, we often need to sample from distributions. *Rejection sampling* is a known textbook method for sampling from density f with intractable direct sampling. The basic method (SRS, Figure 1) constructs an *envelope* Mg that is an upper bound on f, where g is a *proposal distribution* from which we can sample easily. Each time we get a sample from g, we accept or reject it with probability depending on the value of g and f in this point. To guarantee efficiency, a good proposal distribution is a necessary knowledge we need to provide to the sampler. In the absence of such knowledge, we typically resort to a uniform upper bound on f which results in high rejection rates and the method stays in textbooks. What's wrong with a high rejection rate? The reason is that for every point proposed, we need to call f to decide whether this point is accepted. If many points are rejected, then f is called many times with few generated samples. When evaluating f is costly, then we are wasting resources.

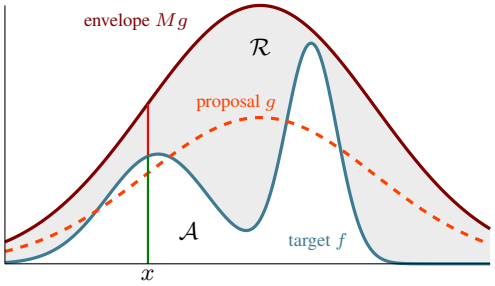


Figure 1. Simple rejection sampling

To alleviate this problem, *adaptive* rejection samplers (Gilks, 1992; Gilks & Wild, 1992; Martino & Míguez, 2011) increase the acceptance rate by taking advantage of particular properties of f. They construct a proposal g that is better adapted to f than just a uniform distribution. Adaptive rejection sampling (ARS, Gilks & Wild, 1992) is the most known among them. ARS works when the target is *log-concave* and constructs a sequence of proposal densities tailored to f. In particular, if a sample that is drawn from a proposal $g_t(x)$ is rejected, this sample is used to build an improved proposal, $g_{t+1}(x)$, with a higher acceptance rate. ARS then adds the rejected point to set S of points defining an envelope of f in order to decrease the area $\mathcal{R}$ between the proposal and the target density (Gilks & Wild, 1992; Gilks, 1992). However, ARS can only be applied for log-concave (and thus *unimodal*) densities, which is a stringent constraint in practice (Gilks & Wild, 1992; Martino & Míguez, 2011) and therefore its use is limited.

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

[1]Research done during author's stay at SequeL, INRIA Lille.

The adaptive rejection Metropolis sampling (Gilks et al., 1995) extends ARS to deal with non-log-concave densities by adding a Metropolis-Hastings (MH) control step after each accepted sample. However, the algorithm produces a Markov chain where the resulting samples are correlated. Another adaptive method is the convex-concave adaptive rejection sampling (Görür & Teh, 2011) where the target distribution is decomposed as the sum of convex and concave functions. In this method, the concave part is treated as in ARS and uses the same set S to construct an upper bound for the convex part by considering the secant lines.

A recent approach is A* sampling (Maddison et al., 2014) that was build on generalizing the Gumbel-Max trick to the continuous case. This method allows to sample from $f(x) \propto \exp(\phi(x))$, where $\phi(x) = i(x) + o(x)$ for some bounded $o(x)$, and some tractable $i(x)$ that is equivalent to the proposal of a classical rejection sampling method. We will relate to A* sampling and compare to it empirically as well. A similar approach is done in OS* (Dymetman et al., 2012) where the sampling is done according to the volume of the region under the proposal.

All these adaptive rejection sampling methods either pose strong assumptions on f or do not come with performance guarantees. In this paper, give an *adaptive strategy* that can work for a general class of densities and *guarantee the number of accepted samples*. An interesting approach for the related, yet different problem of adaptive importance sampling, can be found in the work of Zhang (1996), where the author aims at integrating a function according to a density. To be efficient and sequential, Zhang sequentially approximates the density times the absolute value of the function to be integrated by kernel methods and sample from this approximation. In particular, Zhang estimates the integral of interest by a weighted sum of the collected samples, where the weights depend on the distance between the estimated product function and the true product. This method is interesting because it is non-parametric and therefore requires few assumptions about the shape of the target object.

In this paper, we consider a related idea for rejection sampling. In particular, we use *non-parametric kernel methods* to estimate the target density. This estimate is then used to build a proposal density from which samples are drawn in order to improve the acceptance. This idea is related to the results of Zhang (1996) but there is a significant difference coming from a difference between importance sampling and rejection sampling and which makes the rejection sampling problem harder: While importance sampling requires only a proposal estimator that is good according to the L_2 risk, rejection sampling requires a proposal estimator that is good in L_∞ risk and with high probability. This highlights a fundamental difference between rejection sampling and importance sampling and makes the problem of adaptive rejection sampling significantly more challenging than the problem of adaptive importance sampling.

To address this challenge, we present *pliable rejection sampling* (PRS), a simple variation of rejection sampling. Based on recent advances in density estimation and associated confidence sets, which allow to obtain a *uniform* bound on the estimation error of estimators (Tsybakov, 1998; Korostelev & Nussbaum, 1999; Giné & Nickl, 2010a;b) we propose a method where the proposal is an upper bound on the density that is based on a *kernel estimator* of the density. The motivation behind the choice of a kernel estimator comes (i) from the guarantees on the quality of the estimate and (ii) from the ability to easily sample from it for some specific kernels.

PRS has several advantages. First, it does not pose strong assumptions on f and assumes only mild smoothness properties. For instance, our assumptions are weaker than existing assumptions like log-concavity, concavity or convexity, since if a function satisfies any of these assumptions, then it is in a Besov ball of smoothness two, and therefore smooth enough for our method. Second, it is easy to implement, since it combines common kernel density estimation and traditional rejection sampling. Finally, it comes with a clean and tractable analysis which provides guarantees on the number of samples for a given number of calls to f. Our results imply that asymptotically, if we have a budget of n calls to f, then with high probability, we will obtain n i.i.d samples distributed according to f up to a negligible term. Our procedure is therefore asymptotically almost as efficient as if we were sampling according to f itself.

PRS is actually more efficient than A* sampling in the sense of *budget*. Indeed, in order to generate a single sample from f using A* sampling, we need to consume several calls to f. This implies that even in the asymptotic regime if we have a budget of n calls to f, we will obtain less than $a \times n$ i.i.d samples distributed according to f where $a < 1$ is a small constant. Furthermore, an *huge difference* between PRS and A* sampling, that makes PRS practically appealing, is that the user *does not need to provide* any major information such as a *decomposition* of f into $i(x)$ and $o(x)$ as in the case of A* sampling.

Since PRS is based on rejection sampling, it is useful in the case when the sample space is low-dimensional and when f is not very *peaky*, which is also the case for A* sampling. In particular, A* sampling needs, find the maximum of the convolution of f and a Gumbel process, in order to output a sample. A typical case of a peaky distribution is a posterior distribution commonly present in Bayesian approaches, where computationally efficient MCMC methods (Metropolis & Ulam, 1949; Andrieu et al., 2003) are the tool of choice, as they also scale much better with the dimension. However, the samples are typically correlated

and additional measures need to be taken to make the samples perfect (Andrieu et al., 2003). Even though there exist MCMC methods that perform perfect sampling (Propp & Wilson, 1998; Fill, 1998), they have to assume certain restrictions, and are not used in practice since they are not efficient (Andrieu et al., 2003).

In contrast, our method is a *perfect sampler with high probability*. Our analysis shows that with high probability, asymptotically, each computation of f leads to the sampling of an i.i.d. sample according to f. In Section 3.6, we also provide an extension on how to deal with the high dimensional case and the case of a peaky density (as in the Bayesian posterior case) by a localization method.

2. Setting

Let $d \geq 1$ and let f be a positive function with finite integral defined on $[0, A]^d$ where $A > 0$ (we provide an extension to density defined on $\mathbb{R}^d$ itself in the Appendix B), that we will call the *target density*. Our objective is to provide an algorithm that samples from a normalized version of f with a minimal number of requests to f, where a request is the evaluation of f in a given point of choice. More precisely, the question we ask is the following.

> Given a number n of requests to f, what is the number T of samples $Y_1, \ldots, Y_T$ that one can generate such that they are i.i.d. and sampled according to f?

2.1. Assumption on the target density

We make the following assumptions about f.

Assumption 1 (Assumption on the density)**.** *The positive function f, defined on $[0, A]^d$ is bounded i.e., there exists $c > 0$ such that the density f satisfies $f(x) \leq c$. Moreover, f can be uniformly expanded by a Taylor expansion in any point up to some degree $0 < s \leq 2$, i.e., there exists $c'' > 0$ such that, for any $x \in \mathbb{R}^d$, and for any $u \in \mathbb{R}^d$, we have*

$$|f(x+u) - f(x) - \langle \bigtriangledown f(x), u \rangle \mathbf{1}\{s > 1\}| \leq c'' \|u\|_2^s.$$

For this assumption, we impose that f is defined on $[0, A]^d$, but this could be relaxed to hold for any other convex compact of $\mathbb{R}^d$. For an alternative method and alternative assumptions that do not assume that f has a bounded support, see Appendix B, where this approach is described in detail.

Note that for this assumption, we do not impose that f is a density: it must be a positive function, but it can be a non-normalized density (its integral may not be equal to 1). This remark is particularly useful for Bayesian methods. The assumption also imposes that f is in a Hölder ball of smoothness s. Notice that this is not very restrictive, in particular for a the case with small s.

2.2. Assumption on the kernel

Let K_0 be a positive univariate density kernel defined on $\mathbb{R}$ and let

$$K = \prod_{i=1}^{d} K_0$$

be the d-dimensional product kernel associated with K_0. This kernel will be used in the rest of the paper for interpolating f using collected samples. In order to be able to *sample* from this kernel estimate, it would be more convenient to consider a kernel that corresponds to a density on $\mathbb{R}^d$ (hence a non-negative kernel) from which sampling is easier. A typical example of a useful kernel is then the Gaussian kernel:

$$K_0(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-x^2/2\right).$$

Let us already mention that the Gaussian kernel satisfies also the prerequisites of the next assumption.

Assumption 2 (Assumption on the kernel)**.** *The kernel K_0 defined on $\mathbb{R}$ is uniformly bounded i.e., $K_0(x) \leq C$, and it is a density kernel, i.e., it is non-negative and $\int_{\mathbb{R}^d} K(x)dx = 1$. Furthermore, it is also of degree 2, i.e., it satisfies*

$$\int_{\mathbb{R}} x K_0(x) dx = 0,$$

and, for some $C' > 0$

$$\int_{\mathbb{R}} x^2 K_0(x) dx \leq C'.$$

Also, K_0 is ε-Hölder for some $\varepsilon > 0$, i.e. $\exists\, C'' > 0$ such that for any $(x, y) \in \mathbb{R}^2$,

$$|K_0(y) - K_0(x)| \leq C'' |x - y|^{\varepsilon}.$$

For the Gaussian kernel, the above assumption holds with $C = 1$, $C' = 1$, $C'' = 4$, and $\varepsilon = 1$. In our work, we mainly focus on the case of Gaussian kernel, since it is easy to sample from the resulting estimate, as it is a mixture of Gaussian distributions.

3. Algorithm and results

We first present the main tool which is a kernel estimator and a uniform bound on its performance. We then use it to describe our algorithm, ***p**liable **r**ejection **s**ampling* (PRS). We call our sampler *pliable*, since it builds a proposal by *bending* the original uniform distribution. Moreover, we provide a guarantee on its performance and present extensions to high dimensional situations.

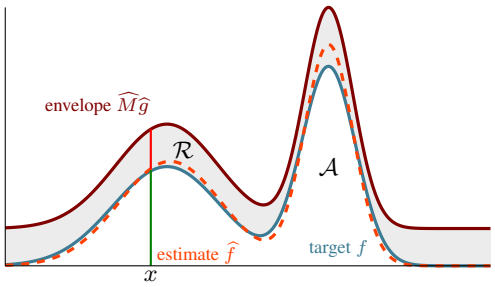


Figure 2. Pliable rejection sampling

3.1. Uniform bounds for kernel regression estimation

Let $X_1, \dots, X_N$ be N points generated uniformly on $[0, A]^d$. Let us define $h \overset{\text{def}}{=} h_s(\delta) = (\log(NA/\delta)/N)^{\frac{1}{2s+1}}$,

$$\widehat{f}(x) = \frac{A^d}{Nh^d} \sum_{k=1}^{N} f(X_i) K\left(\frac{X_i - x}{h}\right). \tag{1}$$

Theorem 1 (proved in Appendix A)**.** *Assume that Assumptions 1 and 2 hold with $0 < s \leq 2$, $C, C', C'', c, c'' > 0$, and $\varepsilon > 0$. The estimate $\widehat{f}$ is such that with probability larger than $1 - \delta$, for any point $x \in [0, A]^d$,*

$$\left|\widehat{f}(x) - f(x)\right| \leq H_0 \left(\frac{\log(NAd/\delta)}{N}\right)^{\frac{s}{2s+d}},$$

where $v = \log\left(1 + \frac{1}{c''+c}\right)\frac{2}{\min(1,s)} + \frac{3}{\varepsilon}\log\left(1 + \frac{1}{C''c}\right)$ and H_0 is a constant that depends on d, v, c, c'', C, C', and A.

3.2. Pliable rejection sampling

PRS (Figure 2, Algorithm 1) aims at sampling as many i.i.d. points distributed according to f as we can with as little computations of f as possible. It consists of three steps:

1. In the beginning PRS samples the domain uniformly at random on $[0, A]^d$ for a number of N samples and computes f for these samples.
2. Then, PRS uses these samples to estimate f by a kernel regression method.
3. Finally, PRS uses the newly obtained estimate plus the uniform bound on it, as a *compact pliable proposal* for rejection sampling.

Since this pliable proposal is close to the target density, the rejection sampling will reject only a small number of points by using it. In PRS, we set the constant

$$N \overset{\text{def}}{=} n^{\frac{2s+d}{3s+d}},$$

Algorithm 1 Pliable rejection sampling (PRS)

Parameters: s, n, δ, H_C
Initial sampling
Draw uniformly at random N samples on $[0, A]^d$
and evaluate f on them
Estimation of f
Estimate f by $\widehat{f}$ on these N samples (Section 3.1)
Generating the samples
Sample $n - N$ samples from
the *compact pliable proposal* $\widehat{g}^\star$
Perform rejection sampling on these samples
using $\widehat{M}$ as a rejection constant to get $\widehat{n}$ samples
Output: Return the $\widehat{n}$ samples

where N is the number of evaluations of the function f needed for the first estimation step that optimizes the number of accepted samples in the second step. We also define

$$r_N \overset{\text{def}}{=} A^d H_C \left(\frac{\log(NAd/\delta)}{N}\right)^{\frac{s}{2s+d}},$$

where H_C is a parameter of the algorithm. Our method samples most of the samples by rejection sampling according to a *pliable proposal* that is defined as

$$\widehat{g}^\star \overset{\text{def}}{=} \frac{1}{\frac{A^d}{N}\sum_{i=1}^{N} f(X_i) + r_N} \left(\widehat{f} + r_N \mathcal{U}_{[0,A]^d}\right), \tag{2}$$

where $\mathcal{U}_{[0,A]^d}$ is the uniform distribution on $[0, A]^d$, and $\widehat{f}$ is the estimate of f defined in (1) computed with the N samples collected in the *initial sampling* phase of PRS. We also define the empirical rejection sampling constant as

$$\widehat{M} \overset{\text{def}}{=} \frac{A^d/N \sum_i f(X_i) + r_N}{A^d/N \sum_i f(X_i) - 5r_N}.$$

3.3. Analysis

In the following, we state and prove the main result about PRS, which is a guarantee on the number of samples.

Theorem 2. *Assume that Assumptions 1, and 2 hold with $0 < s \leq 2$, and that H_C is an upper bound on the constant H_0 from Theorem 1 (applied to f and $\widehat{f}$), and that*

$$8r_N \leq \int_{[0,A]^d} f(x)dx.$$

Then with probability larger than $1 - \delta$, the samples are generated as i.i.d. according to f and for n large enough, the number $\widehat{n}$ of samples generated is at least

$$\widehat{n} \geq n\left[1 - \mathcal{O}\left(\frac{\log(nAd/\delta)}{n}\right)^{\frac{s}{3s+d}}\right].$$

Proof. By Theorem 1 and the definition of r_N, we have that with probability larger than $1-\delta$, for any $x \in [0,A]^d$,

$$\left|\widehat{f}(x) - f(x)\right| \leq r_N \frac{1}{A^d} = r_N \mathcal{U}_{[0,A]^d}.$$

Let ξ' be the event where the above holds. It has probability larger than $1-\delta$. Now let us define event ξ'' as

$$\xi'' \stackrel{\text{def}}{=} \left\{ \left| \frac{A^d}{n} \sum_{i=1}^n f(X_i) - \int_{[0,A]^d} f(x)dx \right| \leq 2A^d c \sqrt{\frac{1}{N}\log(1/\delta)} \stackrel{\text{def}}{=} c_N \right\}.$$

By Hoeffding's inequality, we know that the probability of ξ'' is larger than $1-\delta$. Let $\xi = \xi' \cap \xi''$, the probability of ξ is larger than $1-2\delta$. Therefore, we have that on ξ,

$$\begin{aligned}
\widehat{g}^\star &= \frac{\widehat{f} + r_N \mathcal{U}_{[0,A]^d}}{A^d/n \sum_{i=1}^n f(X_i) + r_N} \\
&\geq \frac{f}{\int_{[0,A]^d} f(x)dx + r_N + c_N} \\
&\geq \frac{f}{\int_{[0,A]^d} f(x)dx}(1 - 4r_N/m),
\end{aligned}$$

with $m = \int_{[0,A]^d} f(x)dx$ and where we used that

$$m \geq 8r_N \geq 4r_N + 4c_N.$$

Note that on ξ

$$\begin{aligned}
\frac{1}{1-4r_N/m} &= \frac{m}{m-4r_N} \\
&\leq \frac{A^d/N \sum_i f(X_i) + c_N}{A^d/N \sum_i f(X_i) - c_n - 4r_N} \\
&\leq \frac{A^d/N \sum_i f(X_i) + r_N}{A^d/N \sum_i f(X_i) - 5r_N} = \widehat{M},
\end{aligned}$$

so that on ξ, the rejection sampling constant $\widehat{M}$ is indeed appropriate. We also have on ξ,

$$\begin{aligned}
\widehat{M} &= \frac{A^d/N \sum_i f(X_i) + r_N}{A^d/N \sum_i f(X_i) - 5r_N} \\
&\leq \frac{m + r_N + c_N}{m - 5r_N - c_N} \\
&\leq \frac{m + 2r_N}{m - 6r_N}.
\end{aligned}$$

Therefore, on ξ, the rejection sampling is going to provide samples that are i.i.d. according to f, and $\widehat{n}$ will be a sum of Bernoulli random variables of parameter larger than

$$\begin{aligned}
\frac{1}{\widehat{M}} &\geq \frac{m - 6r_N}{m + 2r_N} \\
&\geq (1 - 6r_N/m)(1 - 4r_N/m) \\
&\geq 1 - 20r_N/m,
\end{aligned}$$

since $m \geq 8r_N$. We have that on ξ, with probability larger than $1-\delta$,

$$\widehat{n} \geq (n-N)(1 - 20r_N/m) - 2\sqrt{n\log(1/\delta)}.$$

This implies, together with the definition of r_N, $\widehat{n}$ is with probability larger than $1-3\delta$ lower bounded as

$$\widehat{n} \geq (n-N)\left(1 - 20r_N/m - 4\sqrt{\frac{\log(1/\delta)}{n}}\right).$$

Since

$$N = n^{\frac{2s+d}{3s+d}},$$

we have that for n large enough, with probability larger than $1-3\delta$, there exists a constant K such that

$$\widehat{n} \geq n\left[1 - K\log(nAd/\delta)^{\frac{s}{3s+d}} n^{-\frac{s}{3s+d}}\right]. \tag{3}$$

□

Theorem 2 implies that the number of rejected samples is negligible when compared to n: Indeed, the number of rejected samples divided by n is of order

$$\left(\frac{\log(nd/\delta)}{n}\right)^{\frac{s}{3s+d}}.$$

This statement shows a light-years difference between SRS and PRS. Therefore, unlike in SRS where we only accept a fraction of samples, here we asymptotically accept almost all the samples.

3.4. Discussion

Rejected samples. Theorem 2 states that if we have an admissible proposal density g and associated upper bound, as well as a lower bound s on the smoothness of density f, then with high probability, PRS rejects (asymptotically) only a negligible number of samples with respect to n: Almost one sample is generated for every unit of budget spent, i.e., one call of f. This implies in particular that our bounds in terms of a number of i.i.d. samples generated according to f per computation of f are better than the ones for A* sampling (Maddison et al., 2014).

On the other hand, it is not easy to do a direct comparison with MCMC methods since these methods generate correlated samples with stationary distribution f (asymptotically) while we generate exact i.i.d. samples generated according to f (with high probability). However, for any sample generated by MCMC, we need to call f once anyway, which is asymptotically the same as for our strategy.

Sampling from the pliable target. If, for instance, one takes a Gaussian kernel K_0, then sampling from the pliable proposal

$$\widehat{g}^\star = \frac{1}{A^d/N \sum_{i=1}^n f(X_i) + r_N}\left(\widehat{f} + A^d r_N \mathcal{U}_{[0,A]^d}\right)$$

is very easy, since it is a mixture of Gaussian distributions (in $\widehat{f}$, by definition of a kernel estimator), and a uniform distribution on $[0, A]^d$.

The condition on $\int_{[0,A]^d} f(x)dx$. In Theorem 2, we need that

$$\int_{[0,A]^d} f(x)dx \geq 8r_N,$$

so that the empirical rejection sampling $\widehat{M}$ is not too large. If $\int_{[0,A]^d} f(x)dx$ is very small, then it means that f is very peaky and therefore extremely difficult to estimate, besides the trivial case where $f = 0$. This assumption is not very constraining since r_N converges to 0 with N and therefore also with n.

Normalized distribution. If the distribution f is normalized, i.e.,

$$\int_{[0,A]^d} f = 1,$$

then the algorithm can be simplified. Indeed, the *pliable proposal* can be taken as the mixture

$$\frac{1}{1+r_N}\left(\widehat{f} + r_N \mathcal{U}_{[0,A]^d}\right),$$

removing the normalisation constant $A^d/N \sum_i f(X_i)$. In this case, instead of $\widehat{M}$, we can simply use $1 + r_N$ as the rejection sampling constant.

3.5. Case of a distribution with unbounded support

In the case where the distribution f is not assumed to have bounded support, our method does not directly apply since it involves uniform sampling on the domain. One way to go around this, in the case where f is sub-Gaussian, is to sample on uniformly not on $[0, A]^d$, but on a hypercube centered in 0 and of side length $\sqrt{\log(n)}$, and then perform our method using this hypercube as the domain. Then, we would estimate f as 0 outside this hypercube. Because of the properties of sub-Gaussian distributions that have vanishing tails, this will provide results that are similar to the ones on $[0, A]^d$, but with A replaced by $\sqrt{\log(n)}$. Then, for instance, the bound in Theorem 1 provides a bound that would scale on $\mathbb{R}^d$ itself as

$$\left|\widehat{f}(x) - f(x)\right| \leq \mathcal{O}\left(\log(n)^{d/2}\left(\frac{\log(Nd/\delta)}{N}\right)^{\frac{s}{2s+d}}\right),$$

i.e., the bound would become worse by a factor $\log(n)^{d/2}$. This would imply that the bound of Theorem 2 would also become worse by a factor of $\log(n)^{d/2}$. This is not a problem when d is very small. However, even in the case where d is moderately small, this becomes quickly a problem. For this reason, this may not necessarily be a good approach in all cases, for a density with an unbounded support. To deal with this case, a better idea is to do a two-step procedure of rejection sampling, and then estimate f by density estimation instead of regression estimation. (See Appendix B for more details.) In this way, we avoid the problem of paying this additional $\log(n)^d$ in the bound. The algorithm is however slightly more complicated.

3.6. Extensions for high dimensional cases (large d)

One known limitation of rejection sampling is its lack of scalability with the dimension d. While our methodology mainly applies to small dimensions, we now discuss some modifications of the method in order to better handle some specific cases when the ambient dimension d is large, and leverage the scalability of the initial phase. To this end, we resort to optimization techniques that enable to approximately localize the mass of the distribution in time at most quadratic in d (and possibly $\sqrt{d}$), assuming that the density is convex on the region of small mass and arbitrary on the region of high mass:

Definition 1. *We define the γ-support $\text{Supp}_{f,\gamma}$ of f as the closure of its γ-level set $\Lambda_{f,\gamma}$, that is*

$$\text{Supp}_{f,\gamma} = \overline{\Lambda}_{f,\gamma} \quad \text{where} \quad \Lambda_{f,\gamma} \stackrel{\text{def}}{=} \{x \in \mathcal{D} : f(x) > \gamma\},$$

We say one *localizes* the γ-support of f if it finds some $x \in \text{Supp}_{f,\gamma}$. This is however non-trivial:

Lemma 1. *In the general case when no assumption is made on f, localizing the 0-support of f may take a number of evaluation points exponential in the dimension d.*

Proof. Indeed, using uniform sampling, this requires at least $|\mathcal{D}|/|\text{Supp}_{f,0}|$ samples on average. If we introduce R such that $\mathcal{D}$ has the same volume as the Euclidean ball of radius R centered at 0, $B_d(R) \subset \mathbb{R}^d$, and similarly r_0 such that $|\text{Supp}_{f,0}| = |B_d(r_0)|$, this means we need $(R/r_0)^d$ samples on average. □

Thus, without further structure, the initial sampling phase of Algorithm 2 may require exponentially many steps. We thus consider a more specific situation. In practice, for numerical stability, it is important to be able to sample points that are not only in $\text{Supp}_{f,0}$ but also in $\text{Supp}_{f,\gamma}$, for $\gamma > 0$ away from 0. Let r_γ be such that $|\text{Supp}_{f,\gamma}| = |B_d(r_\gamma)|$. We assume that $\text{Supp}_{f,0} = \mathcal{D}$ (and thus $r_0 = R$) but $R/r_\gamma = c_\gamma > 1$, where c_γ is not small, say $c_\gamma \geq 2$, which models a situation when it is easy to localize the 0-support but a priori hard to localize the γ-support.

Now, we assume that the restriction of f on the complement of its γ-support, $f_{|\text{Supp}^c_{f,\gamma}}$ is convex. This situation captures practical situations when the mass of the distribution is localized in a few small subsets of $\mathbb{R}^d$. Note that f does not need to be convex on $\text{Supp}_{f,\gamma}$ and that $\text{Supp}_{f,\gamma}$

can consist of several disjoint connected sets; thus f does not need to be unimodal.

Lemma 2. *Under the previous assumptions, if we can additionally evaluate f and its gradient point-wise, it is possible to find a solution x in $\text{Supp}_{f,\gamma-\varepsilon}$ in no more than $O(d^2/\varepsilon^2)$, that is to localize $\text{Supp}_{f,\gamma}$ in less than an exponential number (with d) of trials, by replacing the uniform sampling scheme in the* initial sampling *phase of PRS with a combination of uniform sampling and convex optimization techniques.*

Proof. The proof is as follows. First, since $r_0 = R$, we can find a point x_0 in $\text{Supp}^c_{f,\gamma}$ in $\mathcal{O}(1)$ trials by uniform sampling. Now, since $f_{|\text{Supp}^c_{f,\gamma}}$ is convex, it is maximal on the boundary of its domain, that is, on $\partial\text{Supp}_{f,\gamma}$. Thus, we use standard optimization techniques to find the maximum of f, starting from x_0: Using the fact that f and its gradient can be evaluated point-wise, the simplest gradient descent scheme (see e.g., Nesterov, 2004, Theorem 3.2.2 with parameter 3.2.10) finds a solution x in $\text{Supp}_{f,\gamma-\varepsilon}$ in no more than $\mathcal{O}(d^2/\varepsilon^2)$ evaluation steps. □

Note that using more refined (but more computationally and memory-wise expensive) methods such as the one from (Nesterov, 2004, 4.2.5 p.187) that relies on point-wise evaluations of the Hessian, one can get a solution $x \in \text{Supp}_{f,\gamma-\varepsilon}$ in no more than $\mathcal{O}(\sqrt{\nu}\ln(\nu/\varepsilon))$ steps, assuming we can build a ν-self concordant barrier function (see Nesterov, 2004 for more explanations regarding such functions). As we are able to build $\mathcal{O}(d)$-self-concordant barrier (and even $(1+\lambda(1))d$-self-concordant barrier, see Bubeck & Eldan, 2015; Hildebrand, 2014, but at the price of a possibly high computational cost), it is then possible to get a solution in only $\mathcal{O}(\sqrt{d}\ln(d/\varepsilon))$ steps. This is another example where one can get an exponential improvement over the general situation.

Now, repeating this procedure T_s times (sample a starting point uniformly at random in $\mathcal{D}$, then optimize$f$ from this starting point), we can get T_s evaluation points in $\text{Supp}_{f,\gamma-\varepsilon}$ in only $\mathcal{O}(T_s d^2/\varepsilon^2)$ and respectively $\mathcal{O}(T_s\sqrt{d}\log(d/\varepsilon))$ steps.

Finally, this naturally extends to cases when $f_{|\text{Supp}^c_{f,\gamma}}$ may not be convex, but $T(f)_{|\text{Supp}^c_{f,\gamma}}$ is convex for some known transformation T, and that $T(f)$, its gradient and its Hessian can all be evaluated point-wise: This is useful in particular when f can only be evaluated up to a normalization constant, as is the case here and often in practice.

4. Numerical experiments

We compared PRS to SRS and A⋆ sampling numerically. In particular, we evaluated the sampling rate, i.e., the proportion of samples that a method gives with respect to the number of evaluation of f. This is equal to the definition of *acceptance rate* for SRS and PRS.

All the experiments were run with $\delta = 0.01$. H_C was set through a cross-validation in order to provide a good proposal quality, i.e., how close is the proposal to the target distribution. H_C is a problem dependent quantity and can capture prior information on the smoothness of f.

The goal of our experiments is to (i) show that PRS outperforms SRS with the same amount of evaluations of f and (ii) that PRS's performance is comparable to A⋆ sampling, which is a recent state-of-the-art sampler, We use two of the same settings in of Maddison et al. (2014). We emphasize again that A⋆ sampling is *given extra information* in form of the decomposition, $f(x) \propto \exp(i(x) + o(x))$ that PRS does not need and that is not available in general.

4.1. Scaling with peakiness

We first study the behavior of the acceptance rate with as a function of to the peakiness of f. In particular, we use the target density of Maddison et al. (2014),

$$f(x) \propto \frac{e^{-x}}{(1+x)^a},$$

where a is the *peakiness parameter*. By varying a, we can control the difficulty of accepting a sample coming from a proposal distribution. For A⋆ sampling, we use the same decomposition of $\phi(x) = i(x) + o(x)$ as Maddison et al. .

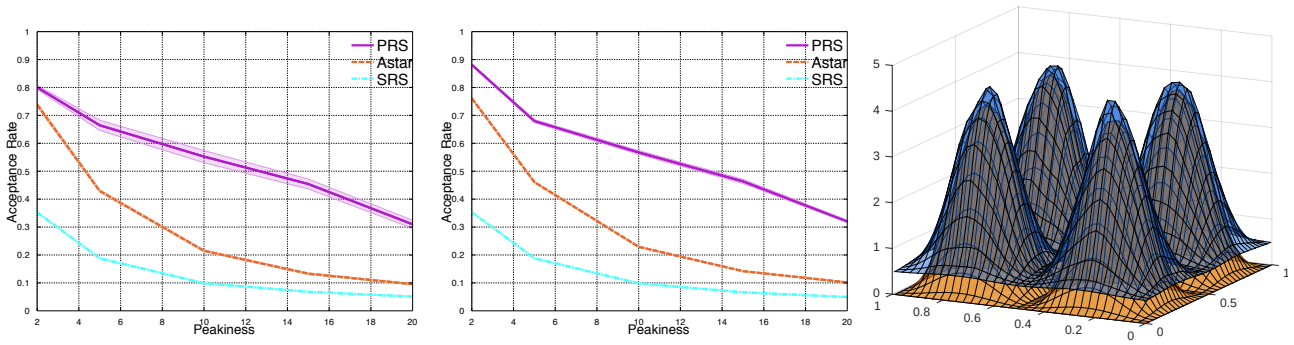
Figure 3 gives the acceptance rates of all these methods for $a \in \{2, 5, 10, 15, 20\}$ averaged over 10 trials. Figure 3a corresponds to a budget of $n = 10^5$ requests to f and Figure 3b to a budget of $n = 10^6$ requests. PRS performs in both cases better than the A⋆ sampling and SRS. Moreover, the performance of PRS improves with n. Indeed, with a larger number of evaluations, the estimate $\widehat{f}$ gets better and more precise, allowing the construction of a tighter upper bound. This provides a good quality proposal that is, in this case, able to perform better than SRS and even outperform A⋆ sampling even for a low peakiness.

4.2. Two-dimensional example

In this part, we compare the three methods on the distribution defined on $[0,1]^2$ as

$$f(x,y) \propto \left(1 + \sin\left(4\pi x - \frac{\pi}{2}\right)\right)\left(1 + \sin\left(4\pi y - \frac{\pi}{2}\right)\right).$$

Figure 3 (right) shows the target density, along with the derived envelope. Table 1 gives the acceptance rates of the three methods for $n = 10^6$, where PRS outperforms SRS and approaches the performance of A⋆ sampling.



(a) $n = 10^4$ (b) $n = 10^5$ (c) 2D example

Figure 3. **Left, center:** Acceptance *vs.* peakiness. **Right:** 2D target (orange) and the pliable proposal (blue)

$n = 10^6$	*acceptance rate*	*standard deviation*
PRS	66.4%	0.45%
A* sampling	76.1%	0.80%
SRS	25.0%	0.01%

Table 1. 2D example: Acceptance rates averaged over 10 trials

$n = 10^5$, 1D	*acceptance rate*	*standard deviation*
PRS	79.5%	0.2%
A* sampling	89.4%	0.8%
SRS	17.6%	0.1%

$n = 10^5$, 2D	*acceptance rate*	*standard deviation*
PRS	51,0%	0.4%
A* sampling	56.1%	0.5%
SRS	2.10^{-3}%	10^{-5}%

Table 2. Clutter problem: Acceptance rates averaged over 10 trials

4.3. Clutter problem

In order to illustrate how PRS behaves for inference tasks, we tested the methods on the clutter problem of Minka (2001) as did Maddison et al. (2014). The goal is to sample from the posterior distribution of the mean of normally distributed data with a fixed isotropic covariance, under the assumption that some points are outliers. The setting is again the same as the one of Maddison et al. (2014): In d dimensions, we generate 20 data points, a half from $[-5, -3]^d$ and another half from $[2, 4]^d$, which provides a bimodal posterior that is very peaky.

Table 2 gives the acceptance rates for the clutter problem in the 1D and 2D cases with a budget of $n = 10^5$ requests to the target f. This target is the posterior distribution of the mean. In this case, even if PRS gives a reasonable acceptance rate, it is not performing better than A* sampling.

5. Conclusion

We propose pliable rejection sampling (PRS), an adaptive rejection sampling method that learns its proposal distribution. While previous work on adaptive rejection sampling aimed at decreasing the area between the proposal and the target by iteratively updating the proposals according to sampling, we learn it using a kernel estimator. We show that PRS outperforms traditional rejection sampling and fares well with recent A* sampling. Our main contribution is a high-probability guarantee on the number of accepted samples using PRS, and a guarantee that only a provably negligible number of samples are rejected with respect to the budget.

Since PRS only estimates the proposal once, a possible algorithmic extension of PRS is to iteratively update the kernel estimate as we gather more samples. While this would result in the same theoretical acceptance guarantee as PRS, the empirical performance is likely to be better.

We have also shown how to improve the scalability of the method to handle moderate dimensions — in high dimensions, one would still suffer from numerical and memory cost. However, under the discussed assumptions, we get a number of steps which is polynomial in d, as opposed to exponential in d. Extending the method to even higher dimension is an interesting research direction.

Acknowledgements We thank Chris Maddison for his code of A* sampling. The research presented in this paper was supported by CPER Nord-Pas de Calais/FEDER DATA Advanced data science and technologies 2015-2020, by French Ministry of Higher Education and Research, Nord-Pas-de-Calais Regional Council, French National Research Agency project ExTra-Learn (n.ANR-14-CE24-0010-01), and by German Research Foundation's Emmy Noether grant MuSyAD (CA 1488/1-1).

References

Andrieu, Christophe, De Freitas, Nando, Doucet, Arnaud, and Jordan, Michael I. An introduction to MCMC for machine learning. *Machine Learning*, 50(1-2):5–43, 2003.

Bubeck, Sébastien and Eldan, Ronen. The entropic barrier: A simple and optimal universal self-concordant barrier. In *Conference on Learning Theory*, 2015.

Dymetman, Marc, Bouchard, Guillaume, and Carter, Simon. The OS* algorithm: A joint approach to exact optimization and sampling. Technical report, http://arxiv.org/abs/1207.0742, 2012.

Fill, James Allen. An interruptible algorithm for perfect sampling via Markov chains. *Annals of Applied Probability*, 8(1):131–162, 1998.

Gilks, W. R. Derivative-free adaptive rejection sampling for Gibbs sampling. *Bayesian Statistics*, 4, 1992.

Gilks, W. R. and Wild, P. Adaptive rejection sampling for Gibbs sampling. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 41(2):337–348, 1992.

Gilks, W. R., Best, N. G., and Tan, K. K. C. Adaptive rejection metropolis sampling within Gibbs sampling. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 44(4):455–472, 1995.

Giné, Evarist and Nickl, Richard. Adaptive estimation of a distribution function and its density in sup-norm loss by wavelet and spline projections. *Bernoulli*, 16(4):1137–1163, 2010a.

Giné, Evarist and Nickl, Richard. Confidence bands in density estimation. *The Annals of Statistics*, 38(2):1122–1170, 2010b.

Görür, Dilan and Teh, Yee Whye. Concave-Convex adaptive rejection sampling. *Journal of Computational and Graphical Statistics*, 2011.

Hildebrand, Roland. Canonical barriers on convex cones. *Mathematics of Operations Research*, 39(3):841–850, 2014.

Korostelev, Alexander and Nussbaum, Michael. The asymptotic minimax constant for sup-norm loss in nonparametric density estimation. *Bernoulli*, 5(6):1099–1118, 1999.

Maddison, Chris J, Tarlow, Daniel, and Minka, Tom. A* sampling. In *Neural Information Processing Systems*, 2014.

Martino, Luca and Míguez, Joaquín. A generalization of the adaptive rejection sampling algorithm. *Statistics and Computing*, 21(4):633–647, 2011.

Metropolis, Nicholas and Ulam, S. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

Minka, Tom. Expectation oropagation for approximate Bayesian inference. In *Uncertainty in Artificial Intelligence*, 2001.

Nesterov, Yurii. *Introductory lectures on convex optimization: A basic course*. Kluwer Academic Publishers, 2004.

Propp, James and Wilson, David. Coupling from the oast: A user's guide. In *Microsurveys in Discrete Probability*, 1998.

Tsybakov, A. B. Pointwise and sup-norm sharp adaptive estimation of functions on the Sobolev classes. *Annals of Statistics*, 26(6):2420–2469, 1998.

Zhang, Ping. Nonparametric importance sampling. *Journal of the American Statistical Association*, 91(435):1245–1253, 1996.