ECOLE NATIONALE DE LA STATISTIQUE
ET DE L'ANALYSE DE L'INFORMATION

PROJET MÉTHODOLOGIQUE

ENSAI 3A

On the efficient computing of sampling policy updates for weighted adaptive importance sampling

rédigé par
Markus Goeswein
Hugo Brunet

January 2nd 2023

Abstract

Portier and Delyon's weighted adaptive importance sampling (wAIS) (6) serves to estimate integrals efficiently. It builds on the concept of adaptive importance sampling and tackles one of its major drawbacks. Namely, AIS bears the burden of potentially bad samples. By introducing weights which allow the algorithm to forget poor samples, (6) are able to demonstrate quicker convergence. What mainly differentiates wAIS from standard importance sampling is that the former approximates a suitable sampling policy by iteratively performing updates to it. In (6), moment equations are used to update the sampling policy. In contrast, we focus on another method which utilizes divergence measures in conjunction with a stochastic gradient descent to converge toward the optimal distribution (for estimating an integral). As divergence measures we consider the Kullback-Leibler and Rényi's alpha divergence. We demonstrate the behavior of our updating algorithm and test it in wAIS. The results in the latter case suggest to use Rényi's alpha divergence when computing many estimates is infeasible and to use the Kullback-Leiber divergence when computing many estimates is possible. The latter may then yield a more accurate result.

1 Introduction

Weighted adaptive importance sampling, abbreviated wAIS, belongs to the class of Monte Carlo methods. The goal of these techniques is to evaluate an integral of the form:

$$\int f \, d\mu = \int \varphi \cdot \pi \, d\mu$$

with f being an integrable function with respect to μ. A well known problem which may occur, is that the data generating process preferably generates values in areas which are not pertinent to the integrand. The resulting estimate suffers from poor variance and may require many samples to yield decent results. Importance sampling resolves this issue by sampling in the important regions of the integrand and subsequently scaling the samples with importance weights. While importance sampling bears the risk of producing much worse results, if done correctly, the importance sampling estimate features a lower variance than the standard Monte Carlo estimate. Our subject of interest, wAIS, is derived from the classic importance sampling and they share the same goal of variance reduction.

The rising interest in importance sampling is owed to its great utility in a wide range of applications such as machine learning or computer graphics. To illustrate, computer graphics heavily rely on efficient computation of integrals to lower the time cost of rendering a scene.

We briefly consider the components of wAIS. Importance sampling is characterized by the following equation:

$$I_\pi(\varphi) = \int \varphi \cdot \pi d\lambda = \int \varphi \times \left(\frac{\pi}{q}\right) \times q \; d\lambda = \mathbb{E}_q[w\varphi]$$

where

- φ: $\mathbb{R}^d \to \mathbb{R}$ such that $\int |\varphi|\pi < +\infty$
- $w = \frac{\pi}{q}$
- π is a density
- q is the density which is proposed for sampling

The resulting estimate is $\frac{1}{n}\sum_{i=1}^{n} w(X_i) \times \varphi(X_i)$ which is given by the law of large numbers.

Given a suitable choice of the sampling policy q, the importance sampling estimate achieves a lower variance than the standard Monte Carlo estimate. For that to be the case, q must be proportional to $|\varphi|\pi$, i.e. $q \propto |\varphi|\pi$ (5), (4).

The dependency of the ideal sampling policy on $|\varphi|\pi$ implies that one cannot sample from them. Here, adaptive importance sampling comes into play.

Adaptive importance sampling (in short: AIS) is adaptive in the sense that it approximates appropriate sampling policies by iteratively updating a proposed sampling policy q. Clearly, q must belong to a family of distributions, from which one can simulate.

We consider Portier and Deylon's wAIS which is characterized by the following equation:

$$q_t \underset{\text{notation}}{\equiv} q_{\theta_t}$$

$$\boxed{I_{(q_i)_{0,T-1}}(\varphi) = \frac{1}{N_T}\sum_{t=1}^{T}\alpha_{T,t}\sum_{i=1}^{n_t}\frac{\varphi(x_{t,i})}{q_{t-1}(x_{t,i})}}$$

(6)

The adaptive nature of the algorithm is expressed via the sampling policy q_{t-1}. Notice, the subscript which indicates its current state. Overall wAIS iterates $t = 1, 2, ..., T$ times.

During iteration t, n_t samples are generated according to q_{t-1}, which are subsequently fed to the algorithm and evaluated. (6) introduces weights, designated by α_t. These weights allow the algorithm to forget earlier stages where the quality of drawn samples was poor. The iteration is concluded by an update of q according to a prespecified updating scheme such that $q_{t-1} \leftarrow q_t$.

(6) employs the generalized methods of moments to update q to demonstrate the asymptotic efficiency of wAIS. Alternative updating schemes based around the Kullback-Leibler Divergence or the sampling variance are not explored. The goal of this paper centers

around examining the asymptotic behaviour of wAIS using the Kullback-Leiber approach and Rényi's alpha divergence in conjunction with a stochastic gradient descent as update regiment. Hence, our final results base themselves on two updating algorithms, one for each divergence measure. Thereby, we extend Portier and Delyon's results on wAIS by examining a computationally cheap, albeit noisy way, to perform updates on the sampling policy.

The appeal of divergence measures in updating the sampling policy lies in the fact that we don't need to know the moment equations of our distribution to be able to update it. It suffices to plug in the densities of our desired distribution and the sampling distribution. The algorithm is able to determine the best approximation of the optimal policy for our sampling distribution irrespective of the latter's parametric family. Properties of the approximation of course depend on the choice of divergence measure. In other terms, it allows to approximate complicated distributions with other ones with fairly low effort.

The methodological section (2) solves two core issues. The first one, is dedicated to deriving the gradient of two divergence measure, namely, the Kullback-Leibler criterion and Rényi's alpha divergence. To arrive at a suitable form of the gradient we apply importance sampling and adaptive importance sampling to the measures. The other issue concerns the design of the optimization algorithm. For this purpose, we recall the fundamental concept of gradient ascent (descent) to subsequently arrive at the well-known stochastic gradient ascent (descent). In the design process, we devise one algorithm based on standard importance sampling and two others based on adaptive importance sampling. Among the adaptive ones, one algorithm has emerged accidentally. Surprisingly, it yields impressive results in the simulations, which is why we decided to include it.

Section 3 discusses some core design principles which guided the code implementation. We also provide some suggestions for further improvements of the code. The remainder of section 3 revolves around simulation studies. Tests of the updating algorithms reveal that the convergence behavior of the adaptive algorithms is superior to the one based on the classic importance sampling. Further, we find that when using the collection of normal distributions as sample and target policy, for various values of α, Rényi's alpha divergence displays more stable behavior and better estimates of the variance than the Kullback-Leibler criterion. Kullback-Leibler, however, provides strong estimates of the mean. For other considered distribution families, Kullback-Leibler outperforms Rényi's alpha divergence. Ultimately, each measure has its merits and their behavior strongly depends on the concrete situation. The simulations of wAIS in conjunction with our updating schemes demonstrate that Rényi's alpha divergence yields decent estimates reliably. Meanwhile, Kullback-Leibler is able to produce very accurate estimates, but it does so rather unreliably. We are also able to confirm (6)'s results that quick updates to the sampling policy as opposed to restrictions on the allocation policy improve the performance of wAIS.

2 Methodology

2.1 Kullback-Leibler Divergence

The first algorithm employs the Kullback-Leibler Likelihood. It is negative KL-Divergence which is given by the following expression:

$$D_{KL}(f\|q_\theta) = \int \log \frac{f}{q_\theta} f d\lambda$$

we will work with the likelihood function as it makes sense in statistics to think with likelihood in mind:

$$L(\theta) = -D_{KL}(f\|q_\theta) = \mathbb{E}_f\left[\log q_\theta f\right]$$

To apply stochastic gradient descent we require the gradient of our optimization criterion: $\nabla_\theta L(\theta)$.

The minimization problem of the Kullback-Leibler divergence between the approximating and target distribution can be easily reformulated as maximization of the Kullback-Leibler Likelihood function. It follows

$$\underset{\theta}{\operatorname{argmin}} D_{KL}(f\|q_\theta) = \underset{\theta}{\operatorname{argmax}} L(\theta)$$

One advantage of using the Kullback-Leibler likelihood function as criterion is that its minimizer remains unchanged by multiplicative constants. This is relevant if the target distributions is only known up to a proportional constant. In that scenario, the absence of the normalization constant does not change the maximization problem.

$$L_{cf}(\theta) = c \times L_f(\theta) - \log c$$

from which we deduce that

$$\operatorname{argmin}_\theta L_{cf}(\theta) = \operatorname{argmin}_\theta L_f(\theta)$$

.

Optimization is done with respect to θ. Hence, we single out the relevant parts dependent on θ.

$$\begin{aligned} L(\theta) &= \int \log \frac{q_\theta}{f} f d\lambda \\ &= \int \left[\, (f \times \log q_\theta) - (f \times \log f)\,\right] d\lambda \\ &= \int (f \times \log q_\theta)\, d\lambda - \underbrace{\int (f \times \log f)\, d\lambda}_{\text{independent of } \theta} \end{aligned}$$

We end up with:

$$\nabla_\theta L(\theta) = \nabla_\theta \int f(u) \log q_\theta(u) du$$

We seek to invert the integral and derivation operator. To do so, we check whether the conditions for derivation under the integral sign apply:

$$f: \begin{array}{ccc} E \times I & \mapsto & \mathbb{R} \\ (x,t) & \mapsto & f(x,t) \end{array}$$

- Existence: $(\forall t \in I)\ \ x \mapsto f(x,t) \in \mathbb{L}^1(E)$
- Derivability: $(\underset{\mu}{\forall} x \in E) t \mapsto f(x,t) \in D^1(I)$
- Dominated Convergence: $\exists \varphi : E \mapsto \mathbb{R}_+ \in m(E),\ \int \varphi d\mu < \infty$

  such that $(\forall t \in I)(\underset{\mu}{\forall} x \in E)\ \ \left|\frac{\partial f}{\partial t}(x,t)\right| \leq \varphi(x)$

In our case we are working with $g(u,\theta) = f(u) \log[\, q(u,\theta)\,]$

One should verify the hypotheses:

- **existence**: $u \mapsto g(u,\theta) \in \mathbb{L}^1(\Omega)$
- **derivability**: we consider

  $$\nabla_\theta g(u,\theta) = \begin{bmatrix} \frac{\partial g}{\partial \theta_1}(u,\theta) \\ \vdots \\ \frac{\partial g}{\partial \theta_p}(u,\theta) \end{bmatrix}$$

  Therefore we can swap derivative andexpectation if $\theta \mapsto g(u,\theta)$ is differentiable for almost all u
- **dominated convergence**: One must find a function ϕ such that $\forall \theta_i$ with $i = 1, ..., p$ $g(u,\theta)$ is dominated by ϕ. Given f and q are densities, they are both bounded which should make finding ϕ not an issue.

Interchanging the integral and derivation operator, we receive the following expression:

$$\begin{aligned} \nabla L(\theta) &= \int \nabla_\theta\ g(u,\theta) du \\ &= \int \nabla_\theta\ [f(u) \times \log q(u,\theta)]\, du \\ &= \int [f(u) \times \nabla_\theta\ \log q(u,\theta)]\, du \end{aligned}$$

With the expression of the gradient being clear, we wish to find an unbiased stochastic version of the integral. We combine the gradient with notions from random generation and importance sampling.

We introduce the sampling distribution q_θ and express our gradient in terms of importance sampling.

$$\begin{aligned} \nabla L(\theta) &= \int [f(u) \times \nabla_\theta\ \log q(u,\theta)]\, du \\ &= \int \nabla_\theta\ \log q(u,\theta) \times \frac{f(u)}{q_\theta(u)} \times q_\theta(u) du \end{aligned}$$

$$\boxed{\nabla L(\theta) = \mathbb{E}_{q_\theta}\left[\left(\frac{f(X)}{q_\theta(X)}\right) \times \nabla_\theta \log q_\theta(X)\right]}$$

The resulting estimator is given by:

$$\begin{aligned}\widehat{\nabla L}(\theta) &\underset{N\to\infty}{\overset{LLN}{\approx}} \frac{1}{N}\sum_{t=1}^{T}\sum_{i=1}^{n_t}\left(\nabla_\theta \log q_{\theta_t}(X_i) \times \frac{f(X_i)}{q_{\theta_t}(X_i)}\right) \\ &= \frac{1}{N}\sum_{t=1}^{T}\sum_{i=1}^{n_t}\omega_{\theta_t}(X_i) \times h_{\theta_t}(X_i)\end{aligned}$$

with:

- $\omega_\theta : x \mapsto \frac{f(x)}{q_\theta(x)}$
- $h_\theta : x \mapsto \nabla_\theta \log q_\theta(x)$

2.2 Normalized Importance Sampling

Often, the target distribution f from which we wish to sample is only known up to a proportional constant. In that scenario, "naive" importance sampling will yield a biased estimate. Normalized importance sampling provides a solution to that issue as demonstrated by the following:

$$\mathbb{E}_f\left[h_\theta(X)\right]$$

$$f(x) = \frac{\varphi(x)}{K} = \frac{\varphi(x)}{\int f(u)du}$$

f is the density and K is the normalizaion constant. As K is unknown, one replaces ω_θ by $W_\theta(x) = \frac{\varphi(x)}{q_\theta(x)}$. This yields:

$$\begin{aligned}\mathbb{E}_{q_\theta}\left[W_\theta(X)h_\theta(X)\right] &= \mathbb{E}_{q_\theta}\left[\tfrac{\varphi(X)}{q_\theta(X)}h_\theta(X)\right] \\ &= \int \frac{\varphi(u)}{q_\theta(u)}h_\theta(u) \times q_\theta(u)du \\ &= \int \varphi(u)h_\theta(u)du \\ &= K\int \underbrace{\frac{\varphi(u)}{K}}_{f(u)} h_\theta(u)du \\ &= K \times \mathbb{E}_f\left[h_\theta(X)\right]\end{aligned}$$

The resulting value is proportional to the unknown constant K. We would like to rid ourselves of K. Fortunately, the following holds:

$$\begin{aligned}\mathbb{E}_{q_\theta}\left[W_\theta(X)\right] &= \mathbb{E}_{q_\theta}\left[\tfrac{\varphi(X)}{q_\theta(X)}\right] \\ &= \int \frac{\varphi(u)}{q_\theta(u)} * q_\theta(u)du \\ &= \int Kf(u)du \\ &= K\int f(u)du \\ &= K\end{aligned}$$

The latter is due to f being a density of which the integral equates to 1. Hence, the normalized importance sampling estimator takes the shape:

$$\nabla L(\theta) \approx \frac{\frac{1}{N}\sum_{t=1}^{n_t}\sum_{i=1}^{N}\omega_{\theta_t}(X_i)h_{\theta_t}(X_i)}{\frac{1}{N}\sum_{t=1}^{n_t}\sum_{i=1}^{n_t}\omega_{\theta_t}(X_i)}$$

with:

- (X_i) sampled according to q_θ
- $h_\theta : x \mapsto \nabla_\theta \log q_\theta(x)$
- $f : x \mapsto \varphi(x)/\int f(u)du$
- $W_\theta : x \mapsto \varphi(x)/q_\theta(x)$

An appropriate sampling policy in the case of normalized importance sampling is given by : $q \propto |\varphi - \int \varphi\pi|\pi$. Again, the issue arises that suitable sampling policies can generally not be sampled from, but must be approximated. In this case, one may resort to normalized AIS. (3) The concept of normalized importance sampling will prove important in the simulation section as (6) recommends the usage of normalized wAIS.

2.3 Gradient Ascent

2.3.1 Classical Gradient Ascent

In this section, we design our optimization algorithm. The goal is to have a stochastic gradient descent. Hence, we shall start by recalling the classic gradient descent. Here, we are actually interested in a gradient ascent as we are looking for the maximum of the Kullback-Leibler criterion.

The goal is to find the maximum of L.
in other terms: $dL(\theta_{max}) = 0$
The gradient is a vector of the same dimension as θ which satisfies the following equation:

$$d_x L(h) = \langle\, \nabla L(x) \mid h \,\rangle$$

$$L(x) \underset{h\to 0}{=} L(x_0) + d_{x_0}L(x - x_0) + o(\|h\|)$$

The main idea is that:

$$\begin{aligned}\underset{x}{\operatorname{argmax}}\, L(x) &\underset{h\to 0}{\approx} \underset{x}{\operatorname{argmax}}\left[\, L(x_0) + d_{x_0}L(h_x)\,\right] \\ &= \underset{h_x}{\operatorname{argmax}}\left[d_{x_0}L(h_x)\,\right] \\ &= \underset{h_x}{\operatorname{argmax}}\,\langle\, \nabla L(x) \mid h_x \,\rangle\end{aligned}$$

We are interested to know the direction h_x which maximises L. It holds that

$$\max_{||h_x||=1}\left\langle \frac{\nabla L(x)}{||\nabla L(x)||} \mid h_x \right\rangle = 1$$

and thus, by using Cauchy-Schwarz inequality, we see that:

$$\underset{||h_x||=1}{\operatorname{argmax}} \langle \nabla L(x) \mid h_x \rangle = +\frac{\nabla L(x)}{||\nabla L(x)||}$$

Hence, the direction of greatest ascent is given by

$$h = +\frac{\nabla L(x)}{||\nabla L(x)||}$$

.

Gradient ascent is an iterative optimization procedure. Therefore, we iterate from an initialization value of our parameter θ_0 and gradually approach θ_{max}, a point where L has a local maximum.

$$\theta_{t+1} \leftarrow \theta_t + \gamma \underbrace{\widehat{\nabla_\theta L}(\theta_t)}_{\frac{1}{N}\sum_{i=1}^{N} \omega(X_i) \times \nabla_\theta \left[h_{\theta_t}(X_i)\right]}$$

2.3.2 Using Importance Sampling to estimate the gradient

Notice the shape of the gradient given in the above box. This corresponds to the gradient of an importance sampling estimate. By replacing h_{θ_t} with $\nabla_\theta \log q_\theta(X_i)$, one receives the gradient of the importance sampling estimate based on the Kullback-Leibler criterion as previously seen in section 2.2. We recall

$$\begin{aligned} \widehat{\nabla L}(\theta) &\approx \frac{1}{N}\sum_{i=1}^{N}\left(\nabla_\theta \log q_\theta(X_i) \times \frac{f(X_i)}{q_0(X_i)}\right) \\ &= \frac{1}{N}\sum_{i=1}^{N} \omega(X_i) \times h_\theta(X_i) \end{aligned}$$

with:

- $\omega : x \mapsto \frac{f(x)}{q_0(x)}$
- $h_\theta : x \mapsto \nabla_\theta \log q_\theta(x)$

By plugging the gradient into the gradient ascent, we receive algorithm 1. To alleviate the notation we henceforth denote $q_{\theta_t} \equiv q_t$.

Algorithm 1 Gradient Ascent - IS

Require:

- Initiate $\theta_0 \in \mathbb{R}^p$
- Initiate η_0 (or η for a fixed step size)
- Choose a sampling distribution q_0
- Choose a small value of ε (i.e $\varepsilon \to 0$)
- Choose a number of maximum iterations: max.iter

Sample $X = (X_i)_{1,N}$ from distribution q_0

for $t \in [\![1, \max.iter]\!]$ **do**

  if $\|\nabla f\| < \varepsilon$ **then**

    Break the loop

  end if

  - Compute $\widehat{\nabla_\theta L}(\theta_t) = \sum_i \omega(X_i) h_\theta(X_i)$
  - $\underbrace{\theta_{[t]} \leftarrow \theta_{[t]} + \eta \nabla L(\theta_t)}_{\theta_{t+1} =}$
  - Update η such that $f(x_{t+1}) > f(x_t)$

end for

2.3.3 Using Adaptive Importance Sampling to estimate the gradient

Intuitively, it makes sense to use the distribution which is the closest to the target distribution at each step. In a slightly modified "adaptive" algorithm, which in fact borrows from the concept of adaptive importance sampling, at each step we sample new observations from the latest distribution q_t and update the gradient formula according to θ_t.

Algorithm 2 Gradient Ascent - Adaptive

Require:

- Initiate $\theta_0 \in \mathbb{R}^p$
- Initiate η_0 (or η for a fixed step size)
- Initiate the sampling distribution q_0
- Choose a small value of ε (i.e $\varepsilon \to 0$)
- Choose a number of maximum iterations : max.iter

for $t \in [\![1, \max.iter]\!]$ **do**

if $\|\nabla f\| < \varepsilon$ **then**

Break the loop

end if

- Sample N_t from distribution q_t
- compute

$$\widehat{\nabla_\theta L}(\theta_t) = \frac{1}{N}\sum_{t=1}^{T}\sum_{i=1}^{n_t} \omega_{\theta_t}(X_i) \times h_{\theta_t}(X_i)$$

$$N = \sum n_t$$
$$\omega_\theta : x \mapsto \tfrac{f(x)}{q_\theta(x)}$$
$$h_\theta : x \mapsto \nabla_\theta \log q_\theta(x)$$

- $\underbrace{\theta_{[t]} \leftarrow \theta_{[t]} + \eta \nabla L(\theta_t)}_{\theta_{t+1}=}$
- Update η such that $f(x_{t+1}) > f(x_t)$
- Update q_t with the most recent value of θ_t

end for

return $\theta_{max.iter}$

2.3.4 A Curious Finding

The following algorithm has resulted from a coding mistake. We discuss some of the mathematical implications in section 3.2.1, where likewise the convergence behavior of said algorithm is assessed. As previously hinted, algorithm 3 yields surprisingly good results. We refer to the corresponding section for details.

Algorithm 3 Gradient Ascent - Adaptive with ω inside the Gradient

identical to algorithm 2 except:

$$\widehat{\nabla_\theta L}(\theta_t) = \sum_{t=1}^{T}\sum_{i=1}^{n_t} \nabla_{\theta_t}\left[\omega_{\theta_t}(X_i) h_{\theta_t}(X_i)\right]$$

$$N = \sum n_t$$
$$\omega_\theta : x \mapsto \tfrac{f(x)}{q_\theta(x)}$$
$$h_\theta : x \mapsto \nabla_\theta \log q_\theta(x)$$

2.3.5 Stochastic Gradient Descent

We recall the empirical gradient from previous sections:

$$\nabla L(\theta) \approx \frac{1}{N}\sum_{i=1}^{N} \omega(X_i) \times h_\theta(X_i)$$

- $\omega : x \mapsto \frac{f(x)}{q_0(x)}$
- $h_\theta : x \mapsto \nabla_\theta \ \log q_\theta(x)$

with the X_i sampled from the distribution q_t

Gradient ascent evaluates the entire gradient at every iteration. A less computationally demanding variation of gradient descent can be found in stochastic gradient descent. Here, the gradient is approximated by computing the gradient only on a random subset, also referred to mini-batch, instead of the entire sample. (2)

Hence, the algorithm is now described as follows:

Algorithm 4 Stochastic Gradient Ascent (SGA)

Require:

- Initiate $\theta_0 \in \mathbb{R}^p$
- Initiate η_0 (or η for a fixed step size)
- Initiate the sampling distribution q_0

new Choose γ, the number of samples drawn at each step

- Choose a small value of ε (i.e $\varepsilon \to 0$)
- Choose a number of maximum iterations : max.iter

for $t \in [\![1, \max.iter]\!]$ **do**
  if $\|\nabla f\| < \varepsilon$ **then**
    Break the loop
  end if
  Sample N_t from distribution q_t

new Select a random subset $I_\gamma(t) \subset [\![1\,,\,N]\!]$ according to the uniform distribution $\mathcal{U}([\![1\,,\,N]\!])$

modified Compute

$$\widehat{\nabla_\theta L}(\theta_t) = \frac{1}{\gamma} \sum_{i \in I_\gamma(t)} \nabla_\theta \left[\omega_\theta(X_i) h_\theta(X_i)\right]$$

$$N = \sum N_t$$
$$\omega_\theta : x \mapsto \tfrac{f(x)}{q_t(x)}$$
$$h_\theta : x \mapsto \log q_t(x)$$

- $\underbrace{\theta_{[t]}}_{\theta_{t+1}=} \leftarrow \theta_{[t]} + \eta \nabla L(\theta_t)$
- Update η such that $f(x_{t+1}) > f(x_t)$
- Update q_t with the most recent value of θ_t

end for

Note that algorithm 4 is based on the adaptive version of gradient descent described in algorithm 2.

2.4 Other Divergence Measures

2.4.1 Rényi's Alpha Divergence

As second divergence criterion, we consider Rényi's Alpha divergence. It is characterized by the following equation:

$$\begin{aligned} R_\alpha(p\|q) &= \tfrac{1}{\alpha-1} \log \mathbb{E}_q \left[\left(\tfrac{p(X)}{q(X)} \right)^\alpha \right] \\ R_\alpha(q_\theta\|f) &= \tfrac{1}{\alpha-1} \log \mathbb{E}_f \left[\left(\tfrac{q_\theta(X)}{f(X)} \right)^\alpha \right] \\ &= \tfrac{1}{\alpha-1} \log \mathbb{E}_{q_0} \left[\left(\tfrac{f(X)}{q_0(X)} \right) \left(\tfrac{q_\theta(X)}{f(X)} \right)^\alpha \right] \end{aligned}$$

We can derive the following expression of the gradient of the Rényi's alpha divergence by applying the differentiation under the integral theorem:

$$\nabla_\theta R_\alpha(q_\theta\|f) = \frac{\nabla_\theta}{\alpha-1} \log \mathbb{E}_{q_0} \left[\left(\frac{f(X)}{q_0(X)} \right) \left(\frac{q_\theta(X)}{f(X)} \right)^\alpha \right]$$

we have:

$$\begin{array}{ccccccccc} \mathbb{R}^p & \longrightarrow & \mathbb{R} & \longrightarrow & \mathbb{R} & \longrightarrow & \mathbb{R} & \longrightarrow & \mathbb{R} \\ \theta & \longmapsto & \frac{q_\theta}{f} & \longmapsto & \left(\frac{q_\theta}{f}\right)^\alpha & \longmapsto & \underbrace{\mathbb{E}_f\left[\left(\frac{q_\theta}{f}\right)^\alpha\right]}_{I(\theta)} & \longmapsto & \log I(\theta) \end{array}$$

using the chain rule we can determine the gradient of Rényi's α-divergence:

$$\begin{aligned} \nabla_\theta R_\alpha(q_\theta\|f) &= \frac{\nabla_\theta \mathbb{E}_f\left[\left(\frac{q_\theta}{f}\right)^\alpha\right]}{\mathbb{E}_f\left[\left(\frac{q_\theta}{f}\right)^\alpha\right]} \\ &\underset{\partial \text{ under } \int}{=} \frac{\mathbb{E}_f\left[\nabla_\theta \left(\frac{q_\theta}{f}\right)^\alpha\right]}{\mathbb{E}_f\left[\left(\frac{q_\theta}{f}\right)^\alpha\right]} \\ &\underset{\frac{dx}{dy}=\frac{dx}{dz}\frac{dz}{dy}}{=} \frac{\mathbb{E}_f\left[\alpha\left(\frac{q_\theta}{f}\right)^{\alpha-1} \nabla_\theta\left[\frac{q_\theta}{f}\right]\right]}{\mathbb{E}_f\left[\left(\frac{q_\theta}{f}\right)^\alpha\right]} \\ \nabla_\theta R_\alpha(q_\theta\|f) &\underset{\text{AIS}}{=} \boxed{\frac{\mathbb{E}_{q_\theta}\left[\alpha\left(\frac{f}{q_\theta}\right)^{1-\alpha} \frac{\nabla_\theta q_\theta}{q_\theta}\right]}{\mathbb{E}_{q_\theta}\left[\left(\frac{f}{q_\theta}\right)^{1-\alpha}\right]}} \end{aligned}$$

By using simple IS sampling from q_0 we get:

$$\nabla_\theta R_\alpha(q_\theta\|f) \underset{\text{IS}q_0}{=} \boxed{\frac{\mathbb{E}_{q_0}\left[\alpha\left(\frac{f}{q_\theta}\right)^{1-\alpha} \frac{\nabla_\theta q_\theta}{q_0}\right]}{\mathbb{E}_{q_0}\left[\left(\frac{f}{q_0}\right)\left(\frac{q_\theta}{f}\right)^\alpha\right]}}$$

From there, using the law of large numbers and the continuity of $(x,y) \mapsto \frac{x}{y}$ on $\mathbb{R} \times \mathbb{R}_+^*$ (because we are dealing with probability densities), we deduce an estimator of $\nabla_\theta R_\alpha(q_\theta\|f)$. This shall be denoted as $\widehat{\nabla R_\alpha}$:

Considering $(x_i)_{1,n}$ samples drawn from q_θ

$$J_N^{[\alpha]}(\text{numerator}) = \frac{\alpha}{N} \sum_{i=1}^N \left(\frac{q_\theta(x_i)}{f(x_i)}\right)^{\alpha-2} \cdot \widehat{\nabla_\theta} \frac{q_\theta}{f}(x_i)$$

$$J_N^{[\alpha]}(\text{numerator}) \underset{N\to\infty}{\longrightarrow} \mathbb{E}_{q_\theta}\left[\alpha\left(\frac{q_\theta}{f}\right)^{\alpha-2} \nabla_\theta\left[\frac{q_\theta}{f}\right]\right]$$

$$J_N^{[\alpha]}(\text{denominator}) = \frac{1}{N} \sum_{i=1}^N \left(\frac{q_\theta(x_i)}{f(x_i)}\right)^{\alpha-1}$$

$$J_N^{[\alpha]}(\text{denominator}) \underset{N\to\infty}{\longrightarrow} \mathbb{E}_{q_\theta}\left[\left(\frac{q_\theta}{f}\right)^{\alpha-1}\right]$$

$$
\begin{aligned}
\widehat{\nabla R_\alpha}(\theta) &= \alpha \sum_{i=1}^{n} \frac{\left(\frac{f(x_i)}{q_\theta(x_i)}\right)^{2-\alpha}}{\sum_k \left(\frac{f(x_i)}{q_\theta(x_i)}\right)^{1-\alpha}} \cdot \widehat{\nabla_\theta} \frac{q_\theta}{f}(x_i) \\
&= \alpha \sum_{i=1}^{n} \frac{\frac{1}{q_\theta(x_i)}\left(\frac{f(x_i)}{q_\theta(x_i)}\right)^{1-\alpha}}{\sum_k \left(\frac{f(x_i)}{q_\theta(x_i)}\right)^{1-\alpha}} \cdot \widehat{\nabla_\theta} q_\theta(x_i) \\
&= \alpha \sum_{i=1}^{n} \frac{\left(\frac{f(x_i)}{q_\theta(x_i)}\right)^{1-\alpha}}{\sum_k \left(\frac{f(x_i)}{q_\theta(x_i)}\right)^{1-\alpha}} \cdot \frac{\widehat{\nabla_\theta} q_\theta(x_i)}{q_\theta(x_i)} \\
&= \alpha \sum_{i=1}^{n} \omega_i \cdot h_i(\theta)
\end{aligned}
$$

with:

- $\omega_i(\theta) \underset{\text{déf}}{\equiv} \frac{\left(\frac{f(x_i)}{q_\theta(x_i)}\right)^{1-\alpha}}{\sum_k \left(\frac{f(x_i)}{q_\theta(x_i)}\right)^{1-\alpha}}$
- $h_i(\theta) \underset{\text{déf}}{\equiv} \frac{\widehat{\nabla_\theta} q_\theta(x_i)}{q_\theta(x_i)} = \widehat{\nabla_\theta} \log q_\theta(x_i)$

using adaptive importance sampling we derive the following estimator :

$$
\boxed{\widehat{\nabla R_\alpha}_{\text{AIS}} = \alpha \cdot \sum_{t=1}^{T} \sum_{i=1}^{n_t} \omega_i(\theta_t) \cdot h_i(\theta_t)}
$$

This time, we are considering a divergence (not the likelihood function). Therefore, we need to perform a gradient **descent** and not ascent:

$$
\theta_{t+1} = \theta_t - \eta \widehat{\nabla R_\alpha}
$$

2.4.2 Amari's Alpha Divergence

One could also consider the Amari's Alpha Divergence using the estimator derived from the following equations:

$$
\begin{aligned}
A_\alpha(p\|q) &\underset{\text{déf}}{\equiv} \frac{1}{\alpha(\alpha-1)}\left[\int p^\alpha q^{1-\alpha} d\lambda \; - 1\right] \\
&= \frac{1}{\alpha(\alpha-1)}\left(\mathbb{E}_q\left[\left(\frac{p(X)}{q(X)}\right)^\alpha\right] - 1\right)
\end{aligned}
$$

therefore we can easily derive the gradient:

$$
\nabla_\theta A_\alpha(q_\theta\|f) = \frac{\nabla_\theta}{\alpha(\alpha-1)} \mathbb{E}_f\left[\left(\frac{q_\theta(X)}{f(X)}\right)^\alpha\right]
$$

Again, we combine previous considerations with the concept of importance sampling to derive the following expression:

$$
\boxed{\nabla_\theta A_\alpha(q_\theta\|f) = \frac{1}{\alpha-1} \mathbb{E}_{q_\theta}\left[\left(\frac{q_\theta}{f}\right)^{\alpha-1} \times \left(\frac{\nabla_\theta q_\theta}{q_\theta}\right)\right]}
$$

From this expression we can derive an estimator of the gradient :

$$
\widehat{\nabla_\theta A_\alpha}(q_\theta\|f) = \frac{1}{N(\alpha-1)} \sum_{t=1}^{T} \sum_{i=1}^{n_t} \left(\frac{q_t(x_i)}{f(x_i)}\right)^{\alpha-1} \frac{\widehat{\nabla_\theta} q_t(x_i)}{q_t(x_i)}
$$

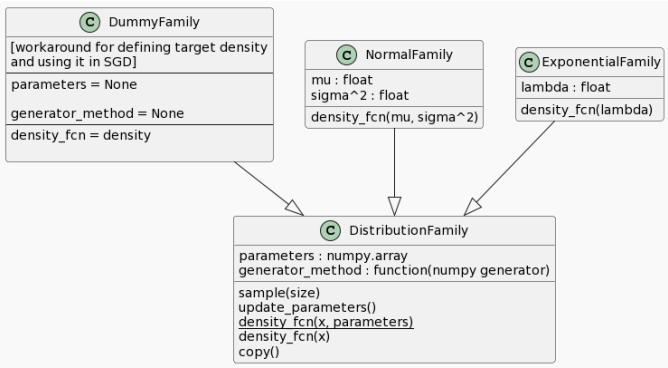
3 Implementation and Simulations

3.1 Implementation

3.1.1 Philosophy of the Code Base

The main idea behind the implementation is to stay as general as possible in order to quickly and easily swap probability distribution families and still have everything working without any further adjustments as long as one can provide a way to sample from that distribution and the expression of the density. In order to achieve this, the code is composed of several functions which operate on instances of a class called DistributionFamily.

A DistributionFamily object is thus composed of methods defining the probability density which can be used in formulas where the density function is used (such as $\widehat{\nabla_\theta L} = \sum_i \omega_i(q_t, f) h_i(q_t, f)$). It is also composed of a sampler method which is used in order to generate samples to evaluate the integrals using importance sampling or adaptive importance sampling.



The code also allows for quick changes of other settings such as the "metrics" used to optimize the parameter of the sampling policy. As long as you can implement the Importance Sampling estimator of the gradient of the likelihood or loss function, you can just pass this estimator (which is a function) as a parameter of the wAIS estimator, and the algorithm will optimize q_t using this criterion instead of having to reimplement the whole algorithm.

As a conclusion, the code was meant to be flexible, in order to approach the optimization of q_t with the best accuracy possible if we have some hints on the form of that distribution. Indeed one could even perform a Gradient Descent using Kullback Leibler or Rényi's alpha-divergence with two distributions from different Families, which allows approximation.

3.1.2 Possible improvements

- Some code hasn't been factorized to its most minimal variant potential. There might be some code that can be reduced in various places. However, the code should be well factorized overall.
- The code doesn't support yet discrete distributions because the gradient descent requires a function which is differentiable, however one could consider using finite difference in order to maybe achieve the same kind of results.
- In order to make sure we still continue to explore the entire space even though we found a distribution which is a local minimum of the Likelihood function, we could consider space exploration tactics such as sampling from another distribution in the same family with a low probability (like 5% and up to 10%). The parameter of that distribution would be determined by going from the stable current parameter and continue going in the same direction as the last gradient step taken.
- Finally one could implement a DistributionFamily child class for mixtures of Normal Laws which could lead to interesting approximations of the target optimal density

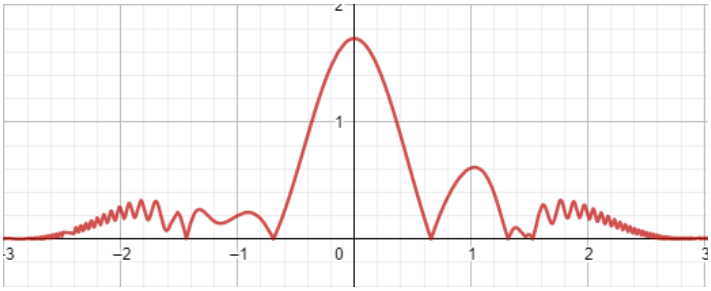


Figure 1: a case of a target optimal policy for wAIS which could be hard to properly approximate using only a normal law

in this particular case :

$$\varphi : x \mapsto \sin(x^5) - 3\cos(3x)$$

$$\pi = \phi : x \mapsto \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}$$

function showed: target density $x \mapsto \pi(x)\,|\varphi(x) - I|$

3.2 Testing the Distribution Convergence

3.2.1 Stochastic Gradient Ascent based on the Kullback-Leibler Criterion

In this section we illustrate the behavior of our algorithms. While their primary purpose is to update the sampling policy in the wAIS framework, we first consider their standalone behavior. We are primarily interested in how the sampling distribution q converges towards the target distribution f.

First, we consider the algorithm based on the Kullback-Leibler criterion. Both, the target density and the sampling policy belong to the family of univariate normal distributions, i.e. $\mathcal{N}(\mu, \sigma^2)$. We fix $f \sim \mathcal{N}(\mu_*, \sigma_*^2)$, initiate $q \sim \mathcal{N}(\mu_0, \sigma_0^2)$ and perform simultaneous updates to μ and σ^2 . During each simulation, that is for each given parametrization of q and f, we employ a stochastic version of algorithm 1, 2 and 3. That is we transfer the ideas from algorithm 4 to algorithm 1 to 3. In an abuse of notation we will likewise refer to the stochastic versions of the algorithms as algorithm 1, algorithm 2 and algorithm 3.

Besides, we set the following parameters:

- N = 80: Number of samples which are generated at each iteration
- γ = 20: Batch size, i.e. subset of N
- η_0 = 0.2: Learning rate
- safety_coeff = 50: Highest admissible value of the gradient norm which circumvents exploding gradients

Parameter configurations are also provided on the figures.

Figure 2 in Appendix A demonstrates the results of the simulation. Namely, we show the relative error curve of the mean (left, i.e. θ_0) and variance (right, i.e. θ_1) for the described setting. Algorithm 1, 2 and 3 are colored in blue, green and orange respectively. Overall, we display three different parametrization, each illustrating a certain type of behavior. They are given by:

- Scenario 1 - high mean and low variance - $f \sim \mathcal{N}(6.32, 1), q_0 \sim \mathcal{N}(13.92, 3.0)$
- Scenario 2 - low mean and high variance - $f \sim \mathcal{N}(-8.34, 1), q_0 \sim \mathcal{N}(1.54, 9.0)$
- Scenario 3 - medium mean and high variance - $f \sim \mathcal{N}(-0.06, 1), q_0 \sim \mathcal{N}(5.42, 20.0)$

The simulations in the scenario of normal target and sampling policy yield the following insights. Algorithm 3 performs the best with respect to the mean, i.e. μ and the variance, i.e. σ^2. Not only does algorithm 3 demonstrate quicker convergence than the other algorithms, but it appears closest to approaching the true value of the parameter in terms of relative error. The standard importance sampling estimate generally lacks behind the other considered algorithms in terms of convergence speed and accuracy. Meanwhile, the adaptive stochastic gradient ascent shows some improvements over algorithm 2. Generally, the estimation of σ^2 poses a greater challenge than that of μ. In the considered scenarios, the relative error of the variance is shown to abandon rather favorable initiation values in favor of worse values. Recovery from these abrupt jumps occurs only at slow to moderate pace or not at all. In some cases it increases even further. s 3 to 6 of appendix B depict the results for other distributions. Again, algorithm 2 and 3 provide superior estimates than algorithm 1 in a breadth of scenarios. Particularly, in the case of normal (fig 2) and logistic (fig 4) distributions, algorithm 3, i.e. $\frac{1}{n}\sum_i \nabla_\theta \left[\left(\frac{f(x_i)}{q_t(x_i)}\right) \cdot \log q_t(x_i)\right]$, seems to gain a lot of speed in convergence. Admittedly, we are not able to provide a mathematical reasoning for the superiority of algorithm 3 and why the algorithm performs especially well in the case of normal and logistic distributions. Algorithm 3 implicitly assumes that the following holds:

$$\begin{aligned}
\nabla_\theta \mathbb{E}_f\left[h(\theta)\right] &= \nabla_\theta \int h(\theta) f \, d\lambda \\
&= \int \nabla_\theta \left[h(\theta) \cdot f\right] \\
&= \int (\frac{q_\theta}{q_\theta}) \cdot f \cdot \nabla_\theta h(\theta) d\lambda = \mathbb{E}_{q_\theta}\left[\left(\frac{f}{q_\theta}\right) \nabla_\theta h(\theta)\right] \\
&= \int \nabla_\theta \left[h(\theta) \cdot \frac{f}{q_\theta} q_\theta\right] d\lambda \\
&\underset{???}{\approx} \int \nabla_\theta \left[h(\theta) \cdot \frac{f}{q_\theta}\right] q_\theta \, d\lambda = \mathbb{E}_{q_\theta}\left[\nabla_\theta \left[h(\theta) \cdot \frac{f}{q_\theta}\right]\right]
\end{aligned}$$

However, we are unable to verify whether the conditions for inverting the derivative and integral apply. There might be certain distribution families which verify the following property:

$$\nabla_\theta \left[h(\theta) \cdot \frac{f}{q_\theta} q_\theta\right] = q_\theta \nabla_\theta \left[h(\theta) \cdot \frac{f}{q_\theta}\right] + \underbrace{\left(h(\theta) \cdot \frac{f}{q_\theta}\right) \nabla_\theta \left[q_\theta\right]}_{\overset{?}{=} o\left(q_\theta \nabla_\theta \left[h(\theta) \cdot \frac{f}{q_\theta}\right]\right)}$$

One may consider deriving an explanation from the fact that both distributions are defined using exponential functions. Even if the logistic distribution does not belong in the exponential family, it might be interesting to look at this rather curious working expression when dealing with the quadratic exponential family as described in Gourieux, Montfort and Trognon (1).

$$f(x|\begin{bmatrix} m \\ \Sigma \end{bmatrix}) = \exp\left[A(m,\Sigma) + B(x) + C(m,\Sigma)x + x^* D(m,\Sigma)x\right] d\lambda$$

Ultimately, we included algorithm 3 in this section for the suprisingly convincing performance. Yet, the algorithm was the product of an accident and lacks mathematical rigor. Hence, we hesitate to recommend its usage. We proceed with algorithm 2 for the testing of Rényi's alpha divergence and the comparison with Kullback-Leibler since it yielded more promising results than algorithm 1.

3.2.2 Stochastic Gradient Descent based on the Renyi Divergence Criterion

Next, we consider the behavior of our algorithm based on Rényi's Alpha Divergence. Previous algorithmic considerations easily apply to Rényi's Alpha Divergence as we are merely required to replace the shape of the gradient with the derivations from section 2.4.1 and employ a gradient descent instead of gradient ascent. Generally, parameter settings of the stochastic gradient descent remain unchanged, except for the safety coefficient. The same holds for the choice of sampling and target distribution. Yet, here we test the convergence behavior once for fixed and once for variable variance. The latter case coincides with our approach for the Kullback-Leibler criterion whereas the former we have not yet examined. For a given parametrization, we illustrate the behavior of the divergence for different values of $\alpha = 0, 2, 5, 30$. Besides, for easy comparison we also depict the behavior of the Kullback-Leibler criterion based on algorithm 2. Figure 6 in appendix B demonstrates the following scenarios:

- Scenario 1 - unknown (i.e. variable) variance - $f \sim \mathcal{N}(7, 5), q_0 \sim \mathcal{N}(15, 9)$
- Scenario 2 - known (i.e. fixed) variance - $f \sim \mathcal{N}(1, 1), q_0 \sim \mathcal{N}(5, 1)$

We summarize the key insights:

- In scenarios where we perform simultaneous updates, i.e. μ and σ^2 are unknown, μ is best approximated by the Kullback-Leibler criterion. However, the approximation of σ^2 via Kullback-Leibler appears inferior to the ones from Rényi's alpha divergence. Particularly, values of $\alpha = 2, 5$ are shown to provide a viable compromise with decent mean and variance convergence.
- Assuming known variance, higher values of α seem to accelerate convergence and diminish noise. In our case, Rényi's alpha divergence with $\alpha = 30$ is unbeaten. Note however, that it only works in the case where we keep the variance fixed. In contrast, the relative error of Kullback-Leibler are more prone to fluctuations. Yet, it does appear to converge in mean, albeit more slowly than other considered options.

However Rényi's divergence criterion might not be suitable for other distribution families as its convergence speed lacks behind Kullback-Leibler's and hence, might not be converging at a satisfying rate. An example of such case can be seen in Appendix C.

3.3 Testing Weighted Adaptive Importance Sampling

Previously we tested the convergence behavior of our updating schemes. Now, we examine how the wAIS fares when using the aforementioned schemes to perform updates to the sampling policy. To maintain some consistency with the original work of Portier and Delyon, (6), we test our algorithms on the same problem, that is the computation of

$$\mu_* = \int x \phi_{\mu_*, \sigma_*^2}(x) dx$$

, with ϕ_{μ_*, σ_*^2} being the probability density of $\mathcal{N}(\mu, \sigma^2)$, $\mu_* = -7$ and $\sigma^2 = 1$. As the problem already implies, we only consider the computation of μ, fixing σ^2 in the process. The sampling policy also stems from the family of normal distributions with initial value $\mu_0 = -3$. We consider the same combination of allocation policies n_t and number of iterations T as (6). However, we already know that the performance greatly benefits from many and quick updates, i.e. large T, while the choice of allocation policy is of lesser importance (6). Besides, in accordance with (6), we only conduct simulations on normalized wAIS, as the unnormalized version does not produce competitive results.

Updates via the stochastic gradient descent do add some flexibility in the update design. Besides the considerable collection of parameters contained in stochastic gradient descent, we may tweak the number of gradient steps at each update as well as the frequency of updates, i.e. whether we perform updates at each T or only at every few iterations. As for the stochastic gradient descent we adopt the same settings as previously. For Rényi's alpha divergence, we ran the simulations with a value of $\alpha = 15$.

The table below summarizes essential parameter values which have been used for testing and the corresponding mean squared error (MSE) which is computed using the results from the simulation. The values in column "method" are to be read in the following way: Method - T-n_t - frequency of updates - gradient steps per update

We shortly explain the different values:

- Method
  - KL: Kullback-Leibler criterion
  - R: Rényi's alpha divergence with $\alpha = 15$
- Allocation policy, i.e. number of drawn samples at each iteration: $n_t = \{2000, 5000, 20000\}$
- Number of iterations $T = 5, 20, 50$
- Frequency of updates: at every/every second/every fourth iteration
- Number of gradient steps at each update: $\{5, 10\}$, $\{3, 6\}$

For each configuration, we performed 30 estimations of μ_*. The resulting estimates are used to compute the MSE which is displayed in the right column of the table. Corresponding visual illustrations are provided in figure 7 of appendix D.

Config	**Method**	**MSE**
0	KL-5-20 000-1-5	4.242587
1	KL-5-20 000-2-10	7.516624
2	KL-20-5 000-4-6	4.707335
3	KL-20-5 000-2-3	4.069143
4	KL-50-2 000-4-6	3.204496
5	KL-50-2 000-2-3	1.623874
6	R-5-20 000-1-5	3.715118
7	R-5-20 000-2-10	3.297377
8	R-20-5 000-4-6	2.383815
9	R-20-5 000-2-3	2.624274
10	R-50-2 000-4-6	2.405894
11	R-50-2 000-2-3	1.990425

The key insights are:

- Irrespective of the used method, the MSE decreases in the number of iterations, which coincides with results from (6).
- Kullback-Leibler (Config 5) provides the best result, yet globally, it produces those less reliably, i.e. suffers from relatively stronger variance.
- Rényi's alpha divergence delivers more consistent results, with MSEs mostly ranging from roughly 3.7 to 2 compared to MSEs for Kullback-Leibler ranging from 7.5 to 1.6.
- Even though using Kullback-Leibler can produce precise results it does suffer from a lack of reliability. Meanwhile Rényi's alpha divergence produces decent but perhaps less precise results more consistently. In situations where only few estimates can be generated, Rényi's estimate may be a safer option than Kullback-Leibler's. If many estimates can be estimated, the chance of having very good estimates with Kullback-Leibler increases, raising its attractivity.

4 Conclusion

The usage of stochastic gradient descent allows estimating integrals efficiently. We explored the usage of two different divergence measures, namely Kullback-Leibler divergence and Rényi's alpha divergence, for updating the sampling policy toward the optimal one (which is clearly not guaranteed to be from the same family as the sampling distribution).

We have tested the behavior of the divergence measures. We conclude that for the Gaussian case with known variance, the Kullback-Leiber divergence converges the quickest but suffers from more noise than Rényi's alpha divergence. Rényi's alpha divergence takes more time to converge but provides a more stable and accurate approximation. Our advice is to use the Kullback-Leibler divergence for quick convergence with minimal steps and to choose Rényi's alpha-divergence, if a few more steps can be afforded. In the case of unknown variance for a Gaussian target distribution, the Kullback-Leibler divergence converges more rapidly toward the mean, yet, the convergence toward the variance is rather poor compared to the that of Rényi's alpha divergence's approximation. We therefore recommend to use Kullback-Leibler divergence if the parameter or integral of interest is more sensitive to the mean evaluation and Rényi's alpha divergence otherwise. These findings specifically apply in the case of normal distributions. If other distributions are considered, as demonstrated, the behavior of the updating algorithms may greatly differ. Hence, which algorithm is more appropriate heavily depends on the concrete situation.

Finally, the simulations of wAIS demonstrate that the Kullback-Leibler divergence criterion can yield accurate estimates albeit less reliable ones than those of Rényi's divergence criterion. The box-plots (Appendix D) and the MSE table nicely illustrate these findings. Our recommendation is to use Kullback-Leibler divergence as criterion if generating a larger number of estimates can be afforded. However, if generating many estimates is not feasible we recommend using Rényi's divergence which reliably provides decent results.

A simulations using normal distributions for f et q_t

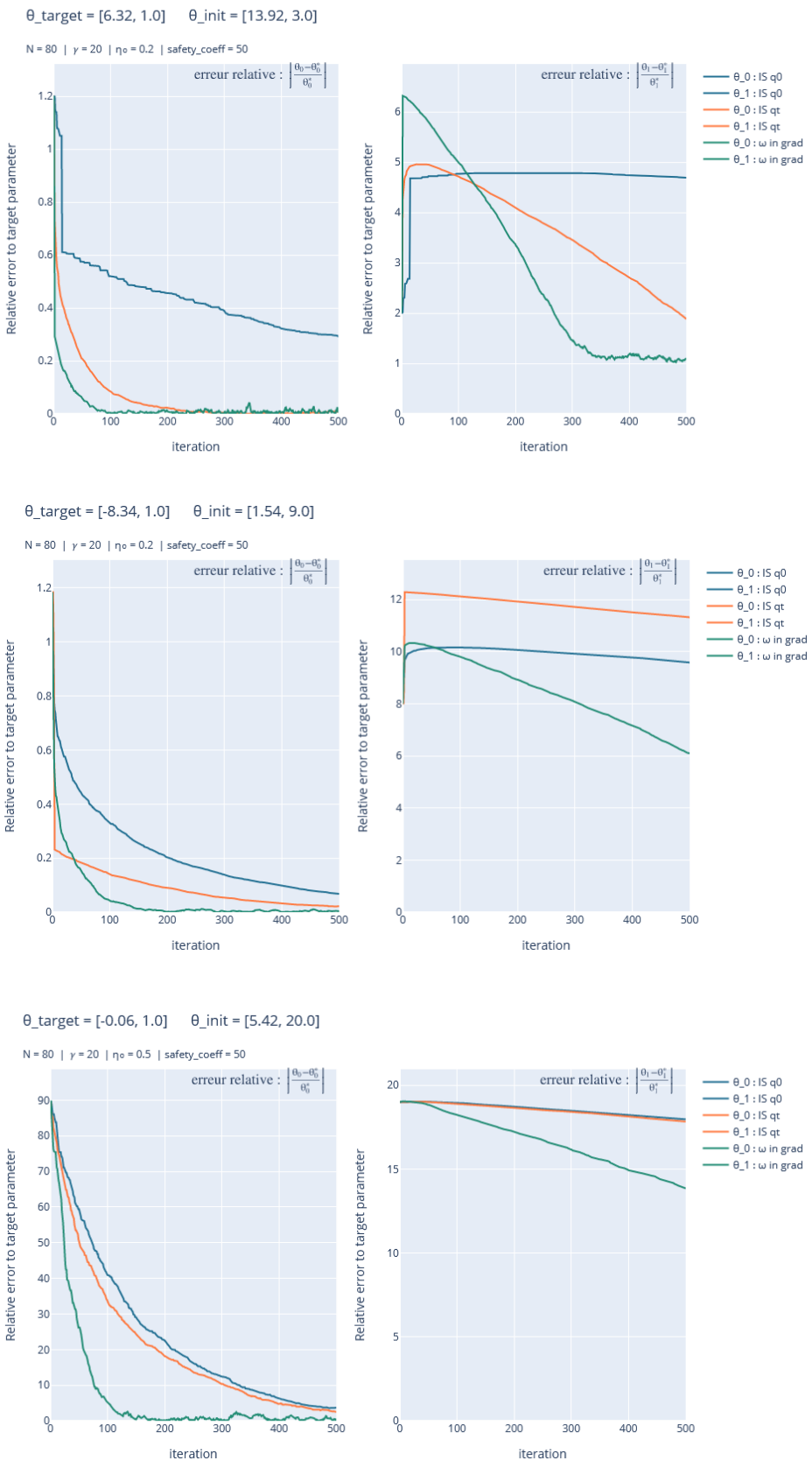


Figure 2: Simulations pour f et q_{θ_t} lois normales

B Other distributions

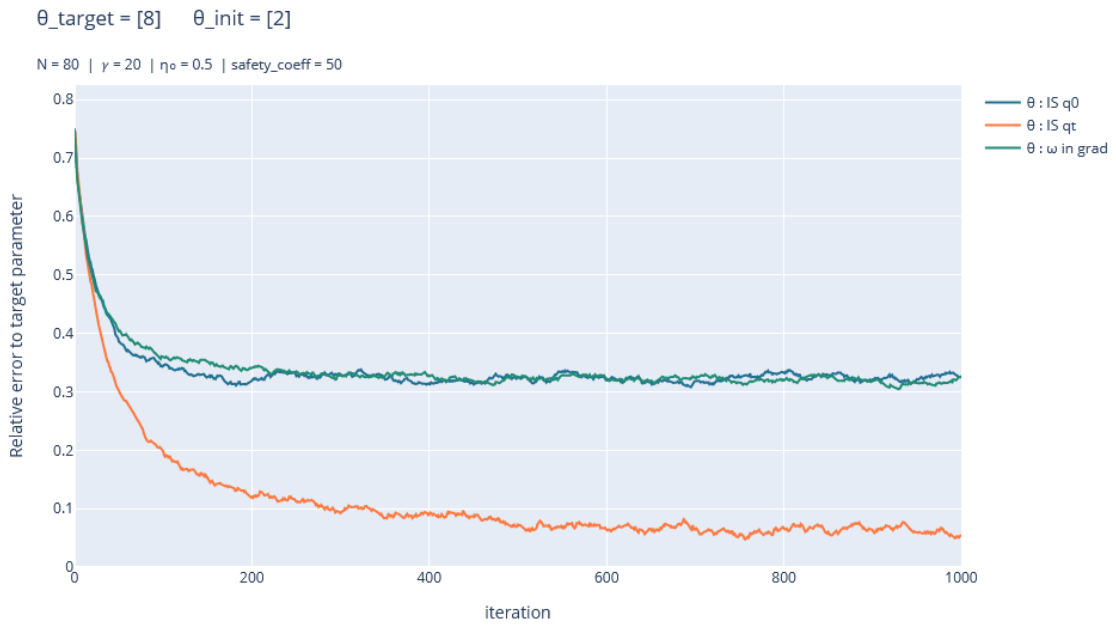


Figure 3: Simulations using Weibull distribution for f and q_{θ_t}

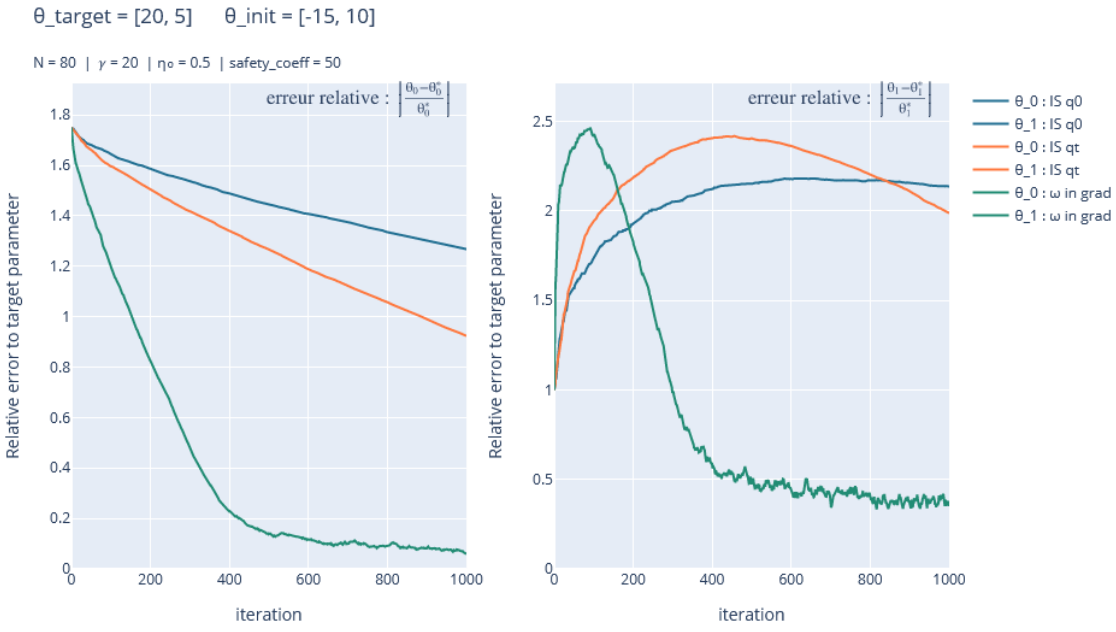


Figure 4: Simulations using logistic distribution for f and q_{θ_t}

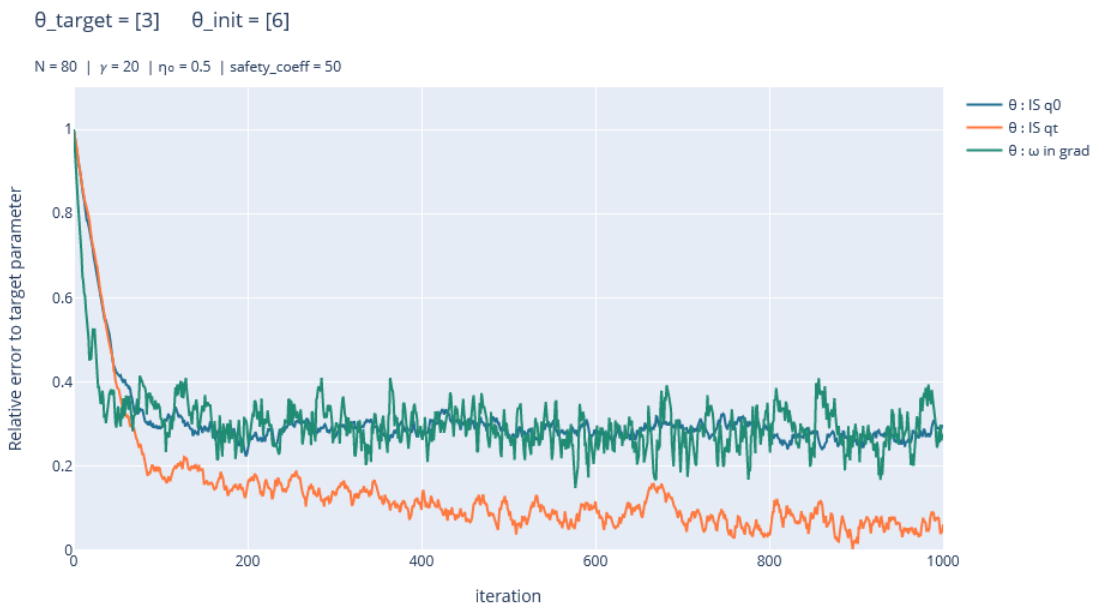


Figure 5: Simulations using exponential distribution for f and q_{θ_t}

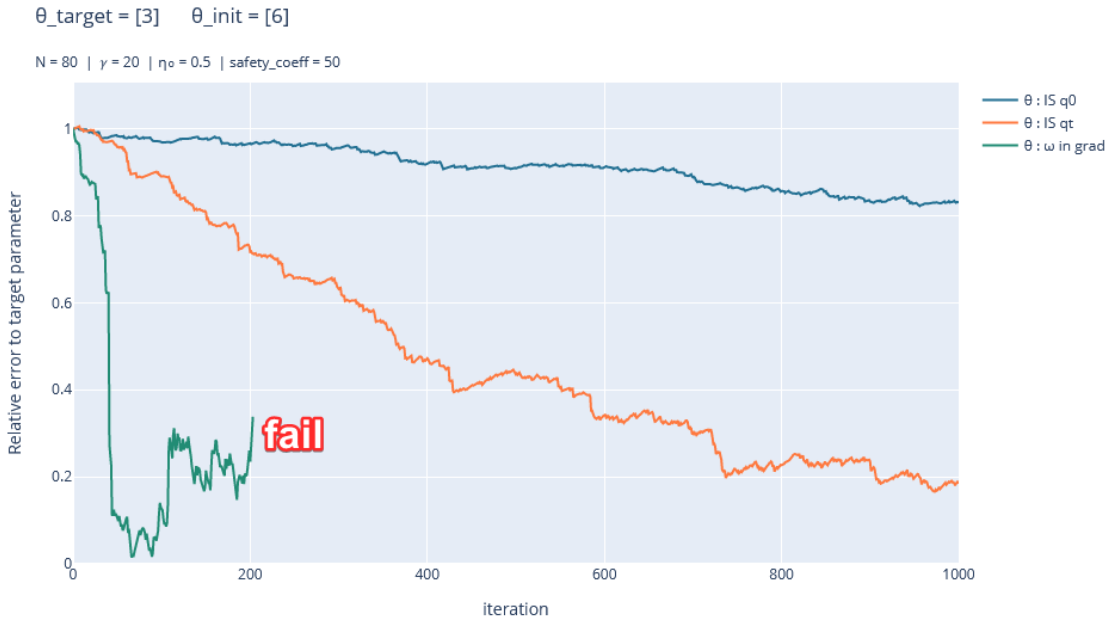
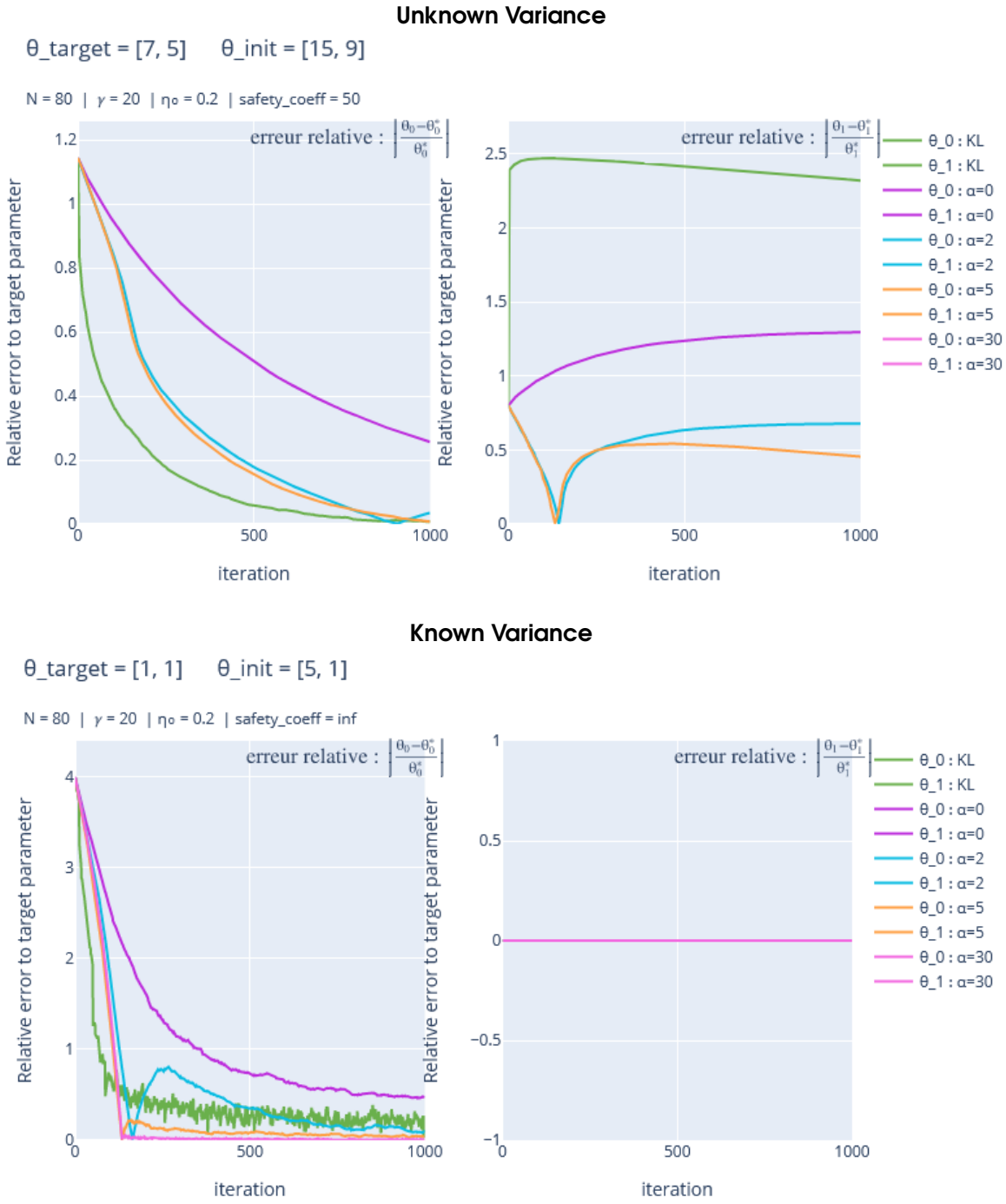


Figure 6: Simulations using student distribution for f and q_{θ_t}

C Rényi's α-Divergence and Kullback-Leibler Divergence



$$\widehat{\nabla R_\alpha}(t) = \alpha \sum_{i=1}^{N} \left[\frac{\left(\frac{f(x_i)}{q_t(x_i)}\right)^{1-\alpha}}{\sum_k \left(\frac{f(x_i)}{q_t(x_i)}\right)^{1-\alpha}} \right] \cdot \frac{\widehat{\nabla}_\theta q_t(x_i)}{q_t(x_i)}$$

$$\widehat{\nabla_\theta L}(\theta_t) = \sum_{i=1}^{N} \underbrace{\frac{f(x_i)}{q_{\theta_t}(x_i)}}_{\omega_t(X_i)} \nabla_\theta \left[h_{\theta_t}(x_i)\right]$$

Figure 7: Simulations of SGA using normal distribution for f and q_{θ_t}: comparing Rényi and Kullback-Leibler Divergence

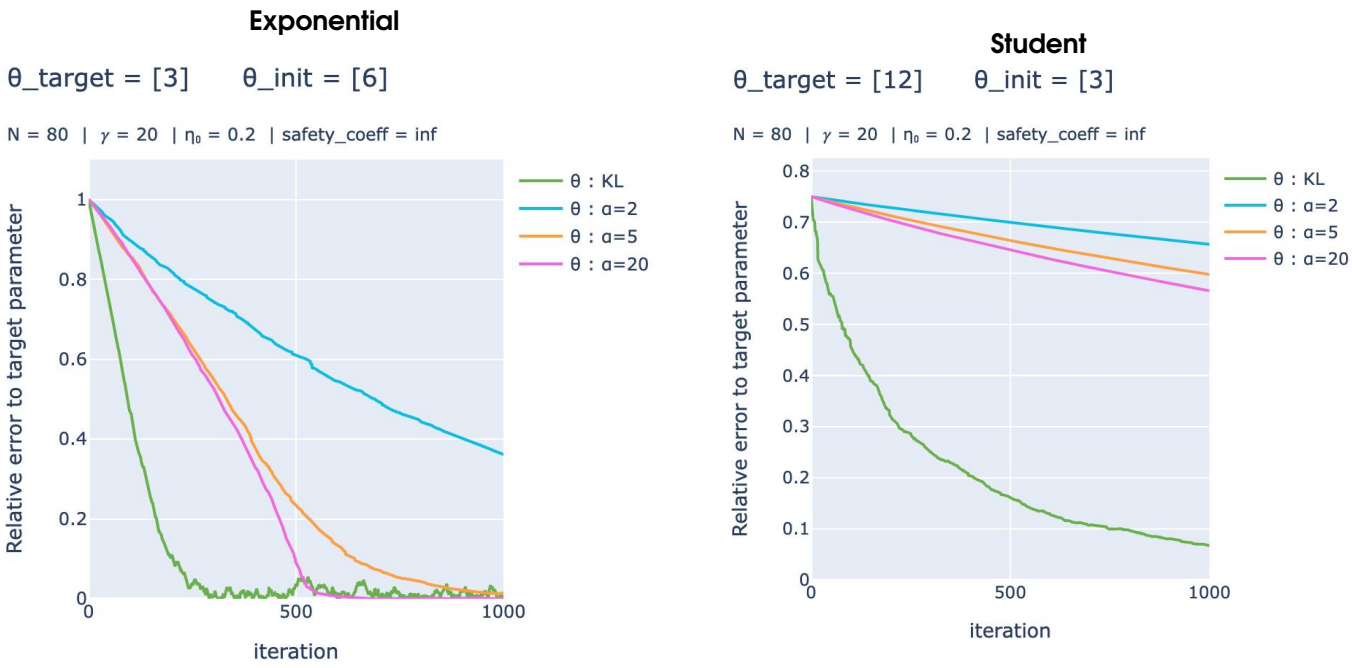


Figure 8: Rényi's divergence criterion for Stochastic Gradient Descent using other distribution families

D wAIS Simulations

Figure 7 depicts the distribution of wAIS estimates, once for the Kullback-Leibler criterion (upper panel) and once for Rényi's alpha divergence with $\alpha = 2$ (lower panel). For every scenario, the distribution is made up of a total of 30 estimates:

- T = 5
  - 30 estimations of μ with update frequency: every iteration (blue)
  - 30 estimations of μ with update frequency: every two iterations (red)
- T = 20 and T = 50
  - 30 estimations of μ using an update frequency: every four iterations (red)
  - 30 estimations of μ using update frequency: every two iterations (green)

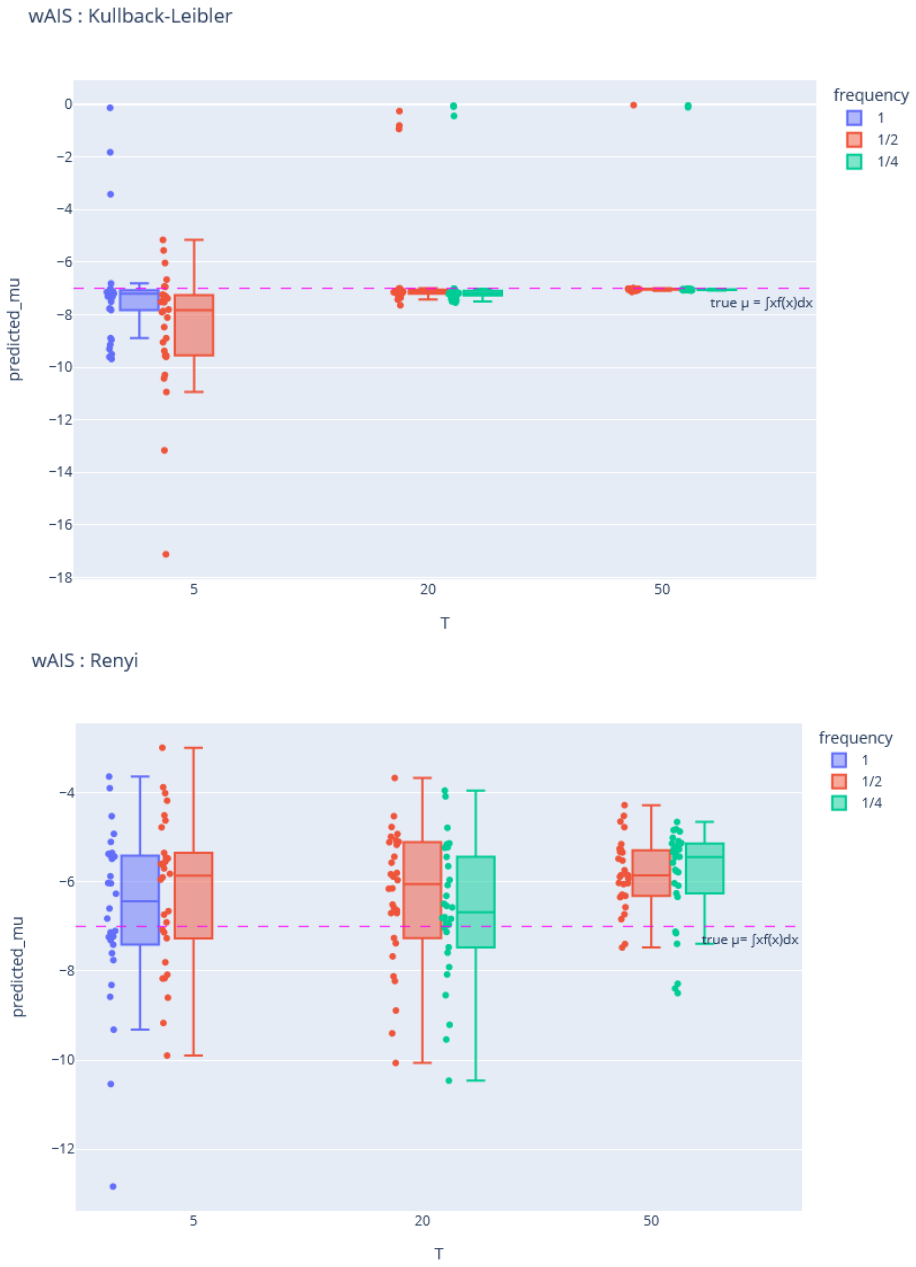


Figure 9: Distribution of the estimates of μ_* = -7 from a normal distribution using wAIS

E Glossary

- **Gradient**: A vector representing the partial derivatives of a function. It represents the direction of steepest slope of a function.
- **Integral**: A function which computes the area under a function curve.
- **Kullback-Leibler Divergence**:

$$L(\theta) = \int \left(\log \frac{q_\theta}{f} \right) f \, d\lambda$$

  The Kullback-Leibler Divergence measures the proximity of two distribution functions q and f, i.e. how similar they are, assuming f is the distribution from which the the sample x_i was generated from. Other divergence measures exist, which measure the similarity differently, i.e. Rényi's alpha divergence.
- **Likelihood function**: A function which outputs a number representing the probability of jointly observing the data under a certain probability function. The higher the likelihood for a given distribution q, the more likely the data has been sampled from q. The likelihood can also be understood as a measure of how plausible the observed data is under a given distribution.
- **Loss function**: The loss function penalizes the deviations of an estimate from the true value it is trying to approximate in a prespecified way.
- **Monte Carlo**: Monte Carlo methods arise from the field of probability theory and serve to approximately solve difficult or infeasible analytical problems via simulations of experiments. An Experiment is a random process with known outcomes, e.g. throwing a dice, that can be performed an infinite amount of times.
- **Normalization constant**: A constant by which we want to divide an integral in order for it to be equal to one and thus making it a probability density function.
- **Probability density function**: A function which characterizes the uncertainty that underlies a random variable. It dictates the patterns in which data appears, i.e. the data generating process, and hence, provides insights into how spread out the data is and where it is centered.
- **Random Variable**: A random variable is a process, event or size whose outcome underlies uncertainty. It is therefore governed by a rule that determines how likely outcomes are to appear. That rule is also referred to as probability density function.
- **Sample**: A collection of values generated from a distribution.
- **Sampling policy**: A probability density function from which samples are generated during the algorithm.
- **Stochastic**: A property of a process or method that it underlies uncertainty and thus relies on probabilities and random events.
- **Weight**: A weight may shift a value, i.e. increase it, decrease it or keep it constant. In the former two cases, this may emphasize or diminish its importance, respectively. In the latter case, i.e. a weight of one, the value remains unchanged by the weight.

F Bibliography

(1) A. Monfort C. Gourieroux and A. Trognon. Pseudo maximum likelihood methods: Theory. *The Econometric Society*, 52(3), 1984. pages 681-700. DOI : https://doi.org/10.2307/1913471.

(2) Faisal Deisenroth and Ong. *Mathematics for Machine Learning*. Cambridge University Press, 2020.

(3) Marin Douc, Guillin and Robert. Minimum variance importance sampling via population monte carlo. *ESAIM: Probability and Statistics*, 11:427–447, 2007.

(4) Evans and Swartz. *Approximating integrals via Monte Carlo and deterministic methods*, volume 20 of *Oxford Statistical Science Series*. Oxford University Press, 2000.

(5) Hammersley and Handscomb. General principles of the monte carlo method. *Monte Carlo Methods*, pages 50–75, 1964.

(6) Portier and Delyon. Asymptotic optimality of adaptive importance sampling. arXiv - 1806.00989, 2018. 7-8.