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

13 décembre 2022

1 Introduction

Do we need an abstract?

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:

$$\int g \, d\mu$$

with g 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. If done correctly, the importance sampling estimate features a lower variance than the standard Monte Carlo estimate. wAIS is a modification of importance sampling, but 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:

$$I_f(g) = \int gfd\lambda = \int g\frac{f}{q}qd\lambda = \mathbb{E}_q[gw]$$

where

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

The resulting estimate is $\frac{1}{n}\sum_{i=1}^{n} w(X_i) \times g(X_i)$ 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 $|g|f$, i.e. $q \propto |g|f$. The dependency $|g|f$ proves trickyFinding such q may prove difficult, which motivates (weighted) adaptive importance sampling.

Portier and Deylon's weighted Adaptive Importance sampling is characterized by the following equation:

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

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

(2)

Instead of struggling to directly fix q optimally, wAIS allows q to approximate the desired sampling distribution in an iterative manner. Hence, 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. Here, Delyon and Portier (2018) introduce 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 $q_{t-1} \leftarrow q_t$.

Delyon and Portier (2018) employ the generalized methods of moments to update q to demonstrate the asymptotic efficiency of wAIS. Alternatives based around the exact methods of moments with the Student distribution, Kullback-Leibler Divergence or the sampling variance are presented, but not further elaborated on. The goal of this paper centers around examining the asymptotic behaviour of wAIS using the Kullback-Leiber approach and (insert criterion).

(insert Structure of the paper) and major results.

2 Methodology

Portier and Deylon's weighted Adaptive Importance sampling is characterized by the following equation:

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

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

(2)

Our project concerns itself with an adequate choice of $q_{t-1}(x_{t,i})$. Notably, $q_{t-1}(x_{t,i})$ is not a fixed value. Rather, in the spirit of the adaptive nature of wAIS, our algorithms are going to iteratively update the sampling policy via a stochastic gradient descent optimization scheme. Precisely, the update is one iteration of the stochastic gradient descent algorithm which yields a cheap and noisy update along the direction of the gradient.

We consider two different algorithms for determining an adequate choice of q. They differ with respect to the criterion they optimize, however, update the sampling policy in an identical manner as described above using the SGD. The first algorithm is based on the Kullback-Leiber loss function, the second algorithm works with techniques which explore a loss landscape.

2.1 Kullback-Leibler Divergence

The first algorithm employs the Kullback-Leibler Loss.

To apply stochastic gradient descent we require the gradient of our optimization criterion. The Kullback-Leibler divergence is defined as

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

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

$$argmin_\theta D_{KL}(f\|q_\theta) = argmax_\theta - D_{KL}(f\|q_\theta)$$

and

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

.

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.

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

We end up with:

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

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

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.

$$\begin{aligned} \nabla L(\theta) &= \int [f(u) \times \nabla_\theta \; \log q(u,\theta)] \, du \\ &= \int \nabla_\theta \; \log q(u,\theta) \times \frac{f(u)}{q_\theta(u)} \times q_\theta(u) du \\ &= \mathbb{E}_{q_\theta}\left[\nabla_\theta \log q_\theta(X) \times \tfrac{f(X)}{q_\theta(X)}\right] \end{aligned}$$

The resulting estimator is given by:

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

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

As previously hinted, 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:

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

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

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:

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

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

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

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:

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

with:

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

2.3 Gradient Ascent

2.3.1 Classical Gradient Ascent

We derived the form of the integral of the respective criterion. This next section will discuss the stochastic gradient descent algorthm. We recall the classic gradient descent. Here, it is presented as gradient ascent as we are looking for the maximum of our function of interest.

The goal is to find the maximum of L.

$$dL(\theta_{max}) = 0$$

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

$$L(x) \approx L(x_0) + d_{x_0} L(x - x_0)$$

$$\begin{aligned} \operatorname*{armax}_x L(x) &\approx \operatorname*{armax}_x \left[\, L(x_0) + d_{x_0} L(h_x) \, \right] \\ &= \operatorname*{armax}_{h_x} \left[d_{x_0} L(h_x) \, \right] \\ &= \operatorname*{armax}_{h_x} \langle \, \nabla L(x) \mid h_x \, \rangle \end{aligned}$$

What is the direction h_x which maximises L?

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

and so:

$$\operatorname*{argmin}_{||h_x||=1} \langle \, \nabla L(x) \mid h_x \, \rangle = + \frac{\nabla L(x)}{||\nabla L(x)||}$$

the direction of greatest ascent is given by

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

.

Therefore, we can iterate from a starting paramater θ_0 and gradually approach to the maximum value of L located at θ_{max} :

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

Gradient Descent computes θ in an iterative manner. To avoid unnessecarily complex notations, we will henceforth call: $q_t \equiv q_{\theta_t}$

Using Importance Sampling according to the distribution q_0

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

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

We can therefore use the following algorithm to compute the gradient ascent toward the optimal parameter θ^*

Algorithm 1 Gradient Ascent - IS

```
Require:
    › Initiate $\theta_0 \in \mathbb{R}^p$
    › Initiate $\eta_0$ (or $\eta$ for a fixed step size)
    › choose a sampling distribution $q_0$
    › choose a small value of $\varepsilon$ ( 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 $\widehat{\nabla_\theta L}(\theta_t) = \sum_i \omega(X_i) h_\theta(X_i)$
        › $\underbrace{\theta_{[t]} \leftarrow}_{\theta_{t+1}=} \theta_{[t]} + \eta \nabla L(\theta_t)$
        › update $\eta$ such that $f(x_{t+1}) > f(x_t)$
  end for
```

It makes sense to use the distribution which is the closet to the one we are interested in at each step. That's why we can also use and "adaptive" algorithm where at each step we sample new observation from the latest distribution q_t. By using this method, one might be careful about the expression of the gradient of the likelihood function. Indeed, as ω will be a function of θ we either need to apply the product rule or compute numerically the gradient of $(\frac{f}{q_t} \times \log q_t)$.

This leads to the following algorithm :

Algorithm 2 Gradient Ascent - Adaptive

```
Require:
    > Initiate $\theta_0 \in \mathbb{R}^p$
    > Initiate $\eta_0$ (or $\eta$ for a fixed step size)
    > Initiate the sampling distribution $q_0$
    > choose a small value of $\varepsilon$ ( 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
          $$\widehat{\nabla_\theta L}(\theta_t) = \sum_{i=1}^{N} \nabla_\theta \left[\omega_\theta(X_i) h_\theta(X_i)\right]$$
              $N = \sum N_t$
              $\omega_\theta : x \mapsto \frac{f(x)}{q_t(x)}$
              $h_\theta : x \mapsto \log q_t(x)$
      > $\underbrace{\theta_{[t]}}_{\theta_{t+1}=} \leftarrow \theta_{[t]} + \eta \nabla L(\theta_t)$
      > update $\eta$ such that $f(x_{t+1}) > f(x_t)$
      > Update $q_t$ with the recently computed parameter
end for
    return $\theta_{max.iter}$
```

2.3.2 Stochastic Gradient Descent

We remember our empirical gradient:

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

› $\omega : x \mapsto \frac{f(x)}{q_0(x)}$

› $h_\theta : x \mapsto \nabla_\theta \ \log q_\theta(x)$

with the X_i sampled from the distribution q_{θ_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 instead of the entire sample. In situations where the data dimension and the data set are very large, evaluating the gradient can be truly expensive. Stochastic Gradient Descent has great merit in such situations.

We have

$$\nabla L(\theta) = \frac{1}{n} \sum_{i \in \text{all obseravtions}} \nabla L_i(\theta)$$

.

Hence the algorithm is now described as follows :

Algorithm 3 Stochastic Gradient Ascent (SGA)

```
Require:
    > Initiate $\theta_0 \in \mathbb{R}^p$
    > Initiate $\eta_0$ (or $\eta$ for a fixed step size)
    > Initiate the sampling distribution $q_0$

  new choose $\gamma$ the number of samples drawn at each step

    > choose a small value of $\varepsilon$ ( 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 $\mathcal{U}\left([\![1\,,\,N]\!]\right)$

      modified compute

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

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

      > $\underbrace{\theta_{[t]}}_{\theta_{t+1}=} \leftarrow \theta_{[t]} + \eta \nabla L(\theta_t)$
      > update $\eta$ such that $f(x_{t+1}) > f(x_t)$
      > Update $q_t$ with the recently computed parameter
  end for
```

2.4 Space Exploration

In order to make sure to explore the space properly, once

2.5 Other divergence measures

2.5.1 Rényi's Alpha Divergence

$$R_\alpha(p\|q) = \frac{1}{\alpha - 1} \log \mathbb{E}_q \left[\left(\frac{p(X)}{q(X)} \right)^\alpha \right]$$

2.5.2 Amari’s Alpha Divergence

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

we can use ideas from importance smapling to derive the following expression :

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

And therefore, we have :

$$A_\alpha(p\|q_\theta) = \frac{\mathbb{E}_{q_0}\left[\omega(X|\theta)\cdot r_\alpha(p\|q_\theta)(X)\right]-1}{\alpha(\alpha-1)}$$

with :

- $r_\alpha(p\|q_\theta) = \left(\frac{p(X)}{q(X)}\right)^\alpha$
- $\omega(X|\theta) = \frac{q_\theta(X)}{q_0(X)}$

The gradient of the divergence can therefore be expressed as (assuming the conditions to invert derivative and integral are met):

$$\nabla_\theta A_\alpha(p\|q_\theta) = \frac{\mathbb{E}_{q_0}\left[\nabla_\theta\left[\omega(X|\theta)\cdot d_\alpha(p\|q_\theta)(X)\right]\right]}{\alpha(\alpha-1)}$$

3 Implementation and Simulations

3.1 Implementation

3.2 Simulations

3.3 Testing Normalized Weighted Importance Sampling

3.3.1 Testing the distribution convergence

The normal (fig 1) and logistic (fig 3) distributions seem to gain a lot of speed in convergence when using $\frac{1}{n}\sum_i \nabla_\theta\left[\left(\frac{f(x_i)}{q_t(x_i)}\right)\cdot\log q_t(x_i)\right]$. We are not sure of the mathematical explanation on why it behaves this way, one might consider to derive a result from the fact both distributions are from the quadratic exponential family as described in Gourieux, Montfort and Trognon (1).

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

3.3.2 Monte Carlo Method for a known Polynomial

We consider a function which has an easy antiderivative to compute such as :

$$P(x) = ax^3+bx^2+cx$$

we decompose the function as $f = h\times q$ to compute it using Mote Carlo methods :

$$\begin{aligned}\int\limits_A^B P(x)dx &= \int\limits_A^B \underbrace{\sqrt{2\pi}e^{\frac{x^2}{2}}P(x)}_{h}\ \underbrace{\frac{e^{\frac{-x^2}{2}}}{\sqrt{2\pi}}}_{q=\mathcal{N}(0,1)}\ dx \\ &= \mathbb{E}_{\mathcal{N}(0,1)}\left[\,h(X)\,\right] \\ &= \sqrt{2\pi}\mathbb{E}_{\mathcal{N}(0,1)}\left[e^{\frac{X^2}{2}}P(X)\right]\end{aligned}$$

However we know the integral precisely :

$$\begin{aligned}\int\limits_A^B P(x)dx &= \left[\tfrac{a}{4}x^4 + \tfrac{b}{3}x^3 + \tfrac{c}{2}x^2\right]_A^B \\ &= \tfrac{a}{4}\left[B^4 - A^4\right] + \tfrac{b}{3}\left[B^3 - A^3\right] \\ &\quad + \tfrac{c}{2}\left[B^2 - A^2\right]\end{aligned}$$

We will first start using a sampling policy $q_0 = \mathcal{N}(\mu_0, \sigma_0)$ and compare the results from the weighted Normalized AIS to the precise evaluation of the integral in order to make sure the algorithm works properly before doing any benchmark.

A simulations using normal distributions for f et q_t

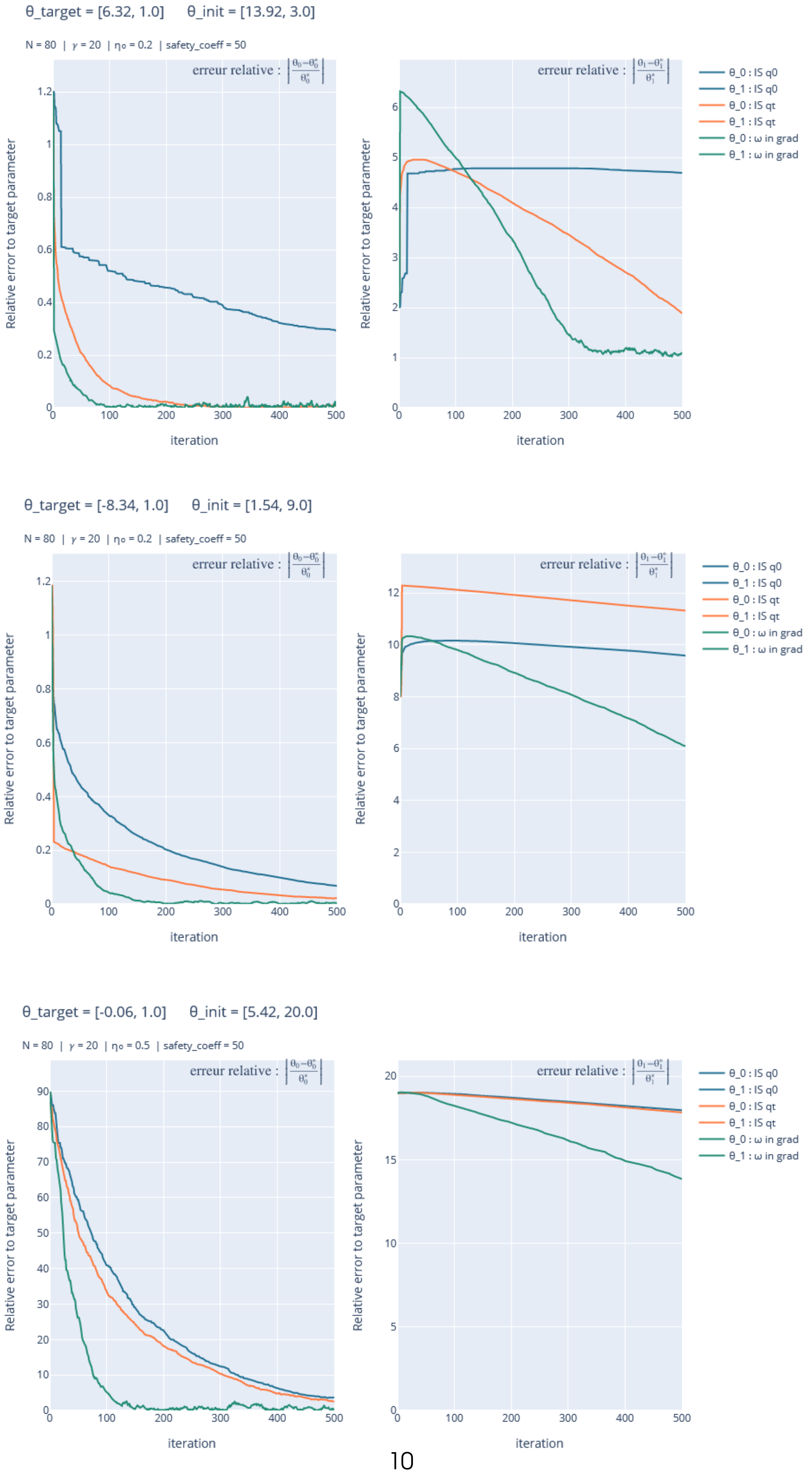

Figure 1: Simulations pour f et q_{θ_t} lois normales

B Other distributions

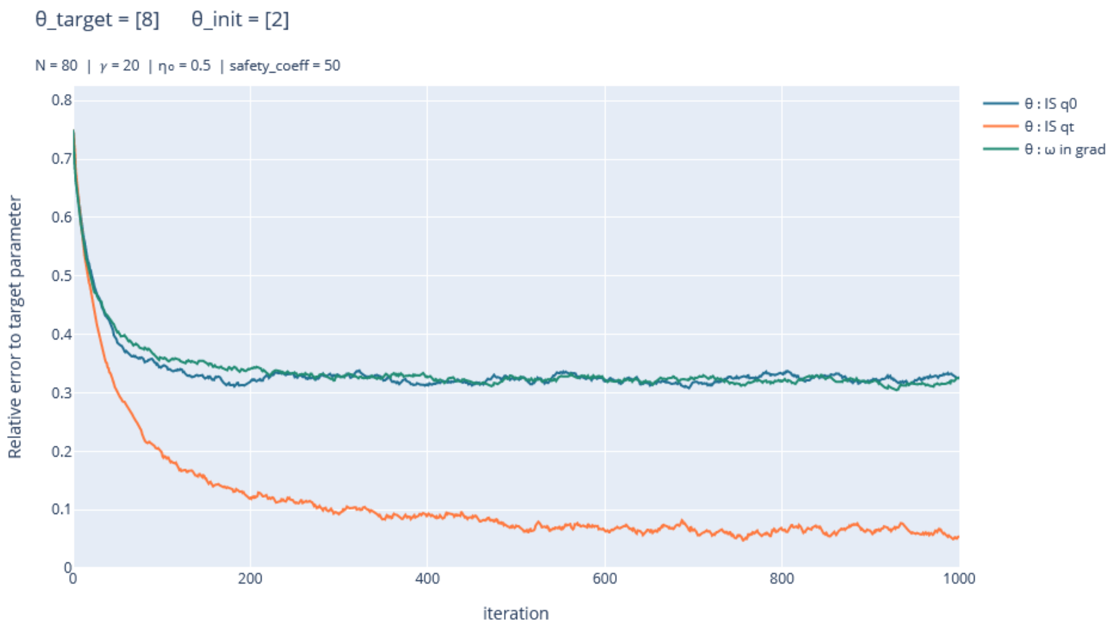

Figure 2: Simulations using Weibull distribution for f and q_{θ_t}

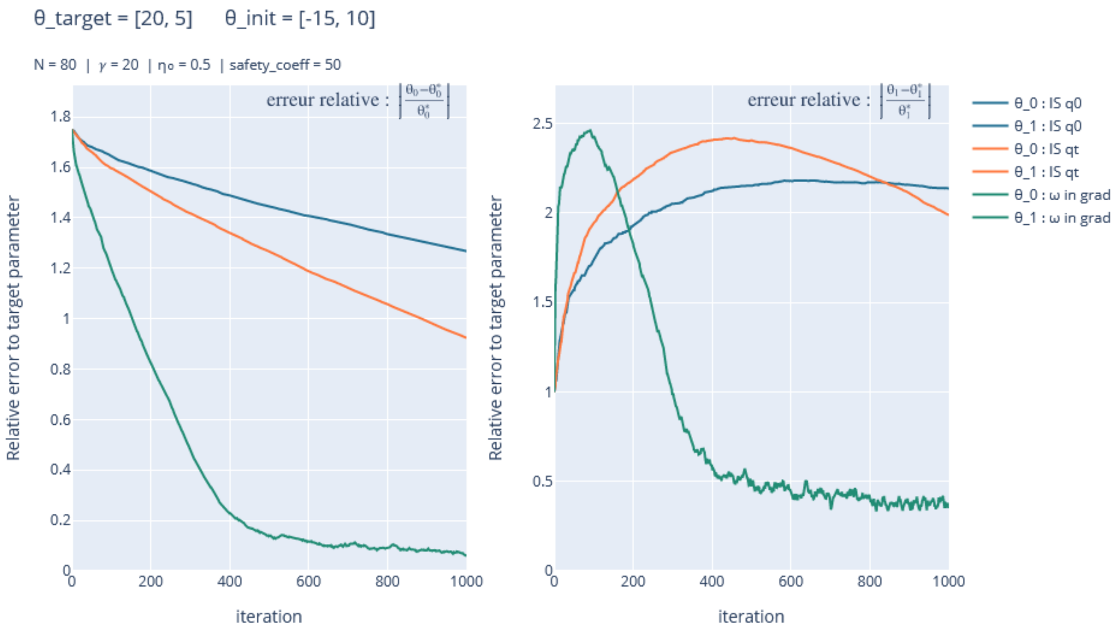

Figure 3: Simulations using logistic distribution for f and q_{θ_t}

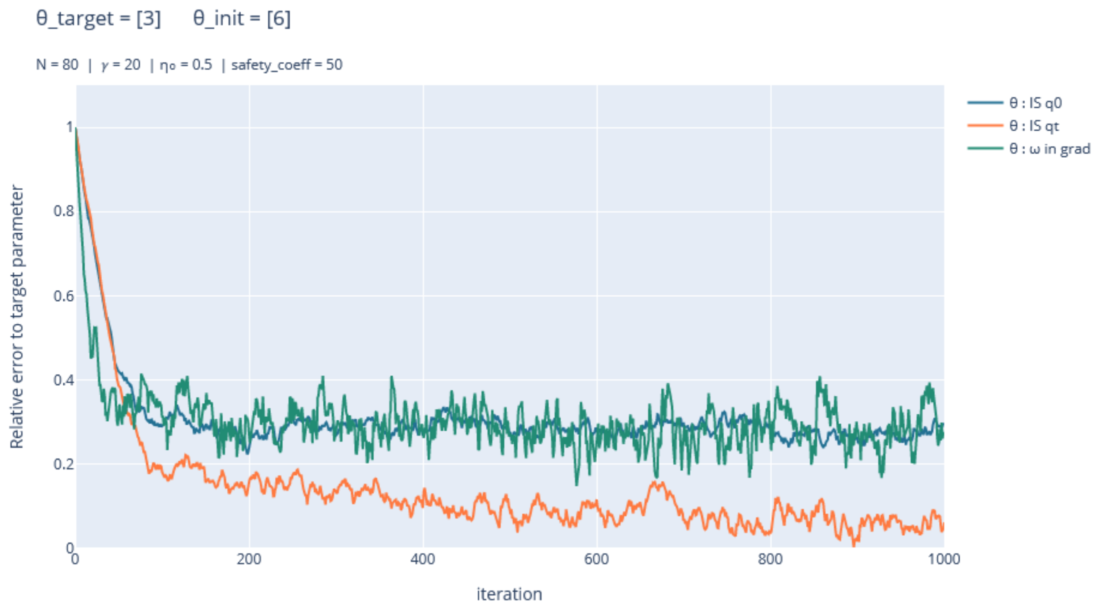

Figure 4: Simulations using exponential distribution for f and q_{θ_t}

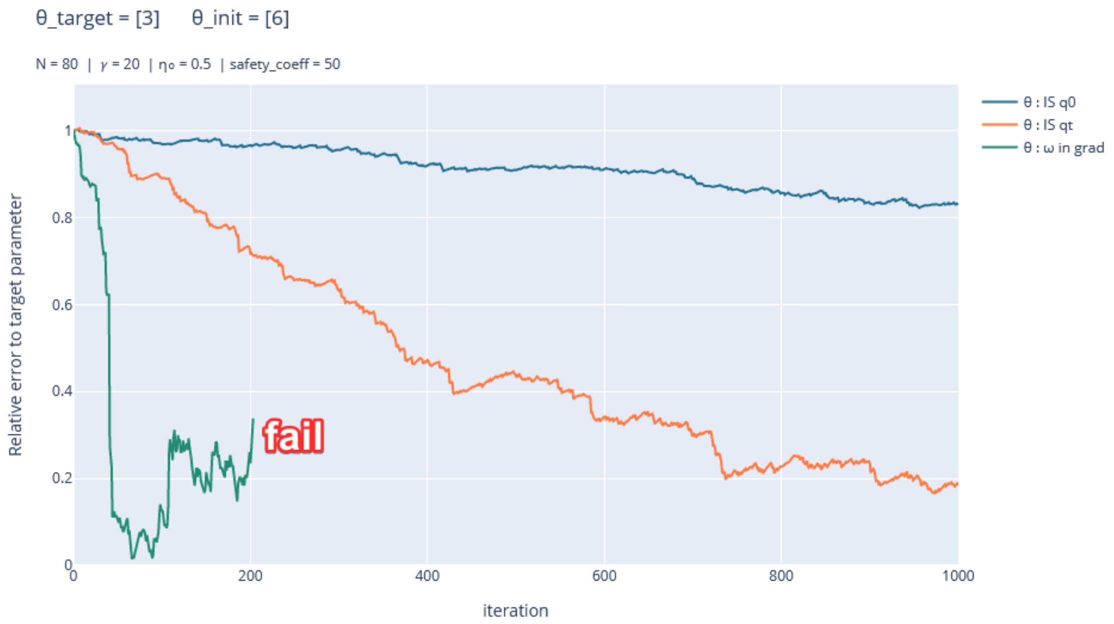

Figure 5: Simulations using student distribution for f and q_{θ_t}

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