Adaptive noisy importance sampling for stochastic optimization

Omer Deniz Akyıldız
Dept. of Signal Theory & Communications
Universidad Carlos III de Madrid
Leganés, Madrid, Spain.
omerdeniz.akyildiz@uc3m.es

Inés P. Mariño
Dept. of Biology & Geology,
Physics & Inorganic Chemistry
Universidad Rey Juan Carlos
Móstoles, Madrid, Spain.
ines.perez@urjc.es

Joaquín Míguez
Dept. of Signal Theory & Communications
Universidad Carlos III de Madrid
Leganés, Madrid, Spain.
joaquin.miguez@uc3m.es

December 2017

Abstract

In this work, we introduce an *adaptive noisy importance sampler* (ANIS) for optimization in an online setting. ANIS is an extension of the family of adaptive importance samplers where the weights are only approximate as they are computed via subsampling of the available data. Allowing errors in the weights enables us to use the algorithm in the so-called large-scale optimization setting, where the cost function consists of the sum of many component functions. ANIS can be used to optimize general cost functions as it does not need any gradient information to update the parameters. We show how the weights of ANIS are related to those of adaptive importance samplers and present some computer simulation results.

1 Introduction

In signal processing and machine learning, the following type of unconstrained optimization problem,

$$\min_{\theta \in \mathbb{R}^d} f(\theta) = \frac{1}{n} \sum_{k=1}^{n} f_k(\theta) \tag{1}$$

where n is large, has attracted a significant attention in recent years. This type of cost function arises in the so-called large-scale data setting, e.g. when it is necessary to fit a model to a large number of data points. For this setting, classical optimization techniques are inefficient because the cost function comprises many individual data points and classical algorithms need to store the whole dataset in order to update parameters at every step. Stochastic (online) optimization algorithms emerged as a powerful alternative, due to their excellent trade-off between statistical and computational efficiency, by obtaining an estimate of the gradient (termed *stochastic gradient*) via subsampling the dataset and taking a stochastic gradient direction at each iteration. This family of algorithms is called stochastic gradient descent (SGD) techniques. The simplest version of the algorithm, as first proposed in [1], computes roots of noisy functions. However, it has been extended in several ways for large-scale machine learning problems, including second-order variants [2] (see [3] for a thorough, recent review of the field). In recent years, there has been considerable research activity in this area and we refer to algorithms of this class as *online optimization* methods in order to emphasize that the randomness is arising from the online selection of data points and also to distinguish them from sampling-based stochastic optimization methods, which are not necessarily online.

Although online extensions of classical optimization methods have become mainstream, online extensions of sampling (simulation) based optimization algorithms have not received much attention. Sampling-based methods aim to sample from a density where the mode coincides with the minimum of the cost function [4]. There are many variants of these methods under different names such as simulated annealing [5]. The advantage of using a population-based algorithm is that it frees the user from computing gradients and removes the differentiability requirement. Thus, general cost functions, for which gradient-based techniques are not straightforward to use, can be optimized using sampling-based techniques. Although sampling-based methods have received a significant interest in the classical optimization literature, they remain relatively unexplored for online optimization problems in the large-scale setting.

There has been some recent work on how to use Markov chain Monte Carlo (MCMC) methods in the online setting. These approaches aim at sampling from a target posterior distribution while using mini-batches at each iteration of the sampler. One line of research relies on the Langevin equation to simulate from the posterior distribution, which requires the computation of gradients [6]. Others follow the classical MCMC approach with noisy Metropolis steps [7, 8]. In principle, one can also use these techniques for stochastic optimization, when the negative log-likelihood is of the form in (1). Our approach, however, is fundamentally different as we extend the family of importance samplers (rather than MCMC methods) to the online setting.

To be specific, we propose a population based online optimizer based on a variant of adaptive importance sampling [9], also known as population Monte Carlo [10]. In particular, we introduce the *adaptive noisy importance sampler* (ANIS) in order to perform optimization in the large scale setting. The ANIS method is a variant of the adaptive importance samplers (AIS), a class of algorithms in which at each iteration the proposal distribution is adapted. However, ANIS is different from conventional AIS algorithms because the weights are noisy, i.e., not exact. The weight error stems from the fact that we use an approximation of the target density rather than the target density itself, a feature that turns out to be useful for running AIS in the large-scale setting. We provide numerical results to show how the ANIS method can be used for optimization when the cost function consists of many terms.

2 Optimization and Bayesian estimation

Before we introduce our algorithm, we briefly look at the intersection between optimization and sampling algorithms. We are interested in solving unconstrained problems of the form,

$$\theta^* = \operatorname*{argmin}_{\theta \in \mathbb{R}^d} f(\theta) \tag{2}$$

where $f(\theta)$ is constructed as in (1). When $f(\theta)$ is differentiable and gradients are *cheap* to compute (i.e., when n is small), this problem can be solved by standard algorithms as simple as the gradient descent. When $f(\theta)$ is differentiable but n is large, then stochastic gradient type algorithms are quite effective to solve this problem. However, when $f(\theta)$ is not differentiable *and* n is large, we need a different approach to find the minimum. To this end, a sampling-based method can be used. In general, sampling algorithms aim at drawing from a target density $\pi(\theta)$ by designing a sampling mechanism, which can be based on importance sampling or MCMC. To see how this can be related to the optimization problem (2), consider a target density $\pi(\theta)$ of the following form,

$$\pi(\theta) = \frac{\exp\left(-\frac{1}{\lambda} f(\theta)\right)}{\int_{\mathbb{R}^d} \exp\left(-\frac{1}{\lambda} f(\theta')\right) \mathrm{d}\theta'} \tag{3}$$

where $f(\theta)$ is a convex function and λ is a scaling constant. In this case, the maximum of the density $\pi(\theta)$ coincides with the minimum of $f(\theta)$ and it can be estimated by drawing samples from $\pi(\theta)$. Moreover, one can obtain the sample mean,

$$\bar{\theta} = \int \theta \pi(\theta) \mathrm{d}\theta.$$

which coincides with the minimum of f when the latter is a symmetric convex function. We are interested in a general convex f in this paper, though. In this case, the mean may be different from the exact minimizer. We further discuss this issue in the experimental section.

The relationship (3) suggests that in order to optimize $f(\theta)$, we can design a sampling mechanism to sample from $\pi(\theta)$. The latter task can be achieved by sampling algorithms such as importance sampling. This is especially useful when $f(\theta)$ is not differentiable or the gradient is not available in closed form.

3 Stochastic optimization with adaptive noisy importance sampling

In this section, we first describe an adaptive importance sampling algorithm (with exact weights) to minimize a given cost function. Then, we introduce the ANIS method, which is an algorithm for stochastic (online) optimization.

3.1 AIS for optimization

Consider a target density of the form in (3). We define a log-concave potential $\phi(\theta) = \exp(-\frac{1}{\lambda} f(\theta))$ where λ is a scaling constant needed for numerical stability. Note that the choice of λ does not change the maximum of $\phi(\theta)$. We also define a proposal density q which is *adaptive*, which means that new samples are drawn conditional on the previous ones. Then $f(\theta)$ can be optimized using a population-based scheme described in Algorithm 1. In this setting, we can set q simply as,

$$q(\theta_t|\theta_{t-1}) = \mathcal{N}(\theta_t; \theta_{t-1}, \sigma^2) \tag{4}$$

for some $\sigma^2 > 0$, where $\mathcal{N}(x; \mu, \sigma^2)$ denotes the Gaussian pdf with mean μ and variance σ^2. We use this proposal in the computer experiments of Section IV.

However, when $f(\theta)$ is of the form in (1), evaluating $\phi(\theta)$ can be quite costly, because one needs to store the whole dataset and evaluate ϕ for each sample used in the population. This renders this algorithm inefficient in the large-scale setting and it is this problem that we tackle in the following section.

3.2 ANIS for online optimization

We have seen that evaluating $\phi(\theta)$ can be very costly when n is large. Modern gradient-based online optimization algorithms use a random mini-batch of data-points, instead of the full cost function, to approximate the gradient. Here, we use the same idea to evaluate importance weights: instead of evaluating weights using the whole cost function, we can obtain *noisy* weights using a random subset of data points.

Instead of evaluating $\phi(\theta)$ at every iteration, we construct a surrogate non-normalized density $\tilde{\phi}_t(\theta)$ at iteration t as follows. We pick a subset of the

Algorithm 1 AIS for optimization

```
Initialise the population $\theta_0^{(i)}$ for $i = 1, \dots, N$.
repeat
    Sample
        $\theta_t^{(i)} \sim \sum_{i=1}^{N} w_{t-1}^{(i)} q(\theta_t | \theta_{t-1}^{(i)})$
    Set
        $W_t^{(i)} = \frac{\phi(\theta_t^{(i)})}{q(\theta_t^{(i)} | \theta_{t-1}^{(i)})}$
    Set
        $w_t^{(i)} = \frac{W_t^{(i)}}{\sum_i W_t^{(i)}}$
    $t \leftarrow t + 1$
until convergence
```

component functions $f_{i_1}(\theta), \dots, f_{i_L}(\theta)$ and construct an estimate of $f(\theta)$ as,

$$\tilde{f}_t(\theta) = \frac{1}{L} \sum_{k=1}^{L} f_{i_k}(\theta).$$

We note that if $i_k \sim U(\{1, \dots, n\})$ are i.i.d and uniform, this estimate of the cost function is unbiased, i.e., we have,

$$\mathbb{E}[\tilde{f}_t(\theta)] = f(\theta).$$

One can alternatively write that $\tilde{f}_t(\theta) = f(\theta) + \xi_t$ where ξ_t is a zero-mean random variable. Moreover, when L is large enough, ξ_t is approximately Gaussian by the central limit theorem. Then we define the non-normalized potential as,

$$\tilde{\phi}_t(\theta) = \exp\left(-\frac{1}{\lambda}\tilde{f}_t(\theta)\right)$$

where λ is a scaling constant and effectively a parameter of the algorithm.

Algorithmically, instead of evaluating $\phi(\theta) = \exp(-\frac{1}{\lambda} f(\theta)$, we evaluate $\tilde{\phi}_t(\theta) = \exp(-\frac{1}{\lambda}\tilde{f}_t(\theta))$ in a *cheap* manner since $L \ll n$. The rest of the algorithm is the same as a conventional AIS. The full procedure is outlined in Algorithm 2.

One can see that the logarithms of noisy weights are unbiased. However, this does not imply that the normalized weights are unbiased as well. In the following, we clarify how the true and the noisy weights are related to each other. Before, we need an assumption.

Assumption 1. *Given that $\tilde{f}_t(\theta) = f(\theta) + \xi_t$, we have*

$$\int \exp\left(-\frac{x}{\lambda}\right) p_{\xi_t}(x) \mathrm{d}x < \infty,$$

Algorithm 2 ANIS for online optimization

```
Initialise the population $\theta_0^{(i)}$ for $i = 1, \dots, N$.
repeat
    Sample
        $\theta_t^{(i)} \sim \sum_{i=1}^{N} \tilde{w}_{t-1}^{(i)} q(\theta_t | \theta_{t-1}^{(i)})$
    Sample $i_1, \dots, i_L \sim \{1, \dots, n\}$ and set
        $\tilde{\phi}_t(\theta) = \exp\left(-\frac{1}{\lambda}\tilde{f}_t(\theta)\right)$
    where $\tilde{f}_t(\theta) = \frac{1}{L}\sum_{k=1}^{L} f_{i_k}(\theta)$.
    Set
        $\tilde{W}_t^{(i)} = \frac{\tilde{\phi}_t(\theta_t^{(i)})}{q(\theta_t^{(i)} | \theta_{t-1}^{(i)})}$
    Set
        $\tilde{w}_t^{(i)} = \frac{\tilde{W}_t^{(i)}}{\sum_i \tilde{W}_t^{(i)}}$
    $t \leftarrow t + 1$
until convergence
```

where $p_{\xi_t}(\cdot)$ is the pdf of the zero-mean error ξ_t.

This assumption means that the expectation of $\exp(-x/\lambda)$ with respect to the probability distribution of the random error ξ_t arising from online selection of the samples must be finite. The reason will be made clear in the following proposition.

Proposition 1. *The expected non-normalized noisy weight for each sample $\theta_t^{(i)}$ is proportional to the non-normalized true weight, i.e.,*

$$\mathbb{E}[\tilde{W}_t^{(i)}] \propto W_t^{(i)}, \quad i = 1, \dots, N.$$

Proof. Recall that we denote the non-normalized "true" weights with $W_t^{(i)} = \phi(\theta_t^{(i)})/q(\theta_t^{(i)}|\theta_{t-1}^{(i)})$. The weights we use in ANIS are computed as

$$\tilde{W}_t^{(i)} = \tilde{\phi}_t(\theta_t^{(i)})/q(\theta_t^{(i)}|\theta_{t-1}^{(i)}),$$

instead. If we compute the expectation of the weights in the ANIS with respect to the distribution of the random error $\xi_t^{(i)}$ introduced by the approximation of

ϕ, we arrive at the following string of equalities,

$$\begin{aligned}
\mathbb{E}[\tilde{W}_t^{(i)}] = \mathbb{E}\left[\frac{\tilde{\phi}_t(\theta_t^{(i)})}{q(\theta_t^{(i)}|\theta_{t-1}^{(i)})}\right] &= \frac{\mathbb{E}\left[\tilde{\phi}_t(\theta_t^{(i)})\right]}{q(\theta_t^{(i)}|\theta_{t-1}^{(i)})} \\
&= \frac{\mathbb{E}\left[\exp\left(-\tilde{f}_t(\theta_t^{(i)})/\lambda\right)\right]}{q(\theta_t^{(i)}|\theta_{t-1}^{(i)})} \\
&= \frac{\exp\left(-\frac{1}{\lambda}f(\theta_t^{(i)})\right)\mathbb{E}\left[\exp\left(-\frac{1}{\lambda}\xi_t^{(i)}\right)\right]}{q(\theta_t^{(i)}|\theta_{t-1}^{(i)})} \\
&= W_t^{(i)}\mathbb{E}\left[\exp\left(-\frac{1}{\lambda}\xi_t^{(i)}\right)\right].
\end{aligned}$$

However, from Assumption 1,

$$\mathbb{E}[\exp(-\xi_t^{(i)}/\lambda)] = \int \exp\left(-\frac{\xi_t^{(i)}}{\lambda}\right)p_{\xi_t}(\xi_t^{(i)})d\xi_t^{(i)} < \infty$$

i.e., $c = \mathbb{E}[\exp(-\xi_t^{(i)}/\lambda)]$ is a finite constant and we arrive at

$$\mathbb{E}[\tilde{W}_t^{(i)}] \propto W_t^{(i)}.$$

■

Corollary 1. *If L is large enough to ensure that $\xi \sim \mathcal{N}(0, v^2)$ for some $v^2 < \infty$ by the central limit theorem, then we have*

$$\mathbb{E}[\tilde{W}_t^{(i)}] = W_t^{(i)}\exp\left(\frac{v^2}{2\lambda^2}\right).$$

These results do not show that the normalized weights are unbiased, which is the requirement for a noisy AIS algorithm. Instead, we have shown that the expected values of the non-normalized noisy weights are proportional to their exact counterparts.

4 Computer experiments

In this section, we present two computer experiments. To evaluate the ANIS method, we use the mean of the distribution $\int \theta\pi(\mathrm{d}\theta) \approx \sum_{i=1}^N \tilde{w}_t^{(i)}\theta_t^{(i)}$ where t is the number of the current iteration – although this choice may not always converge to the true maximum. Alternatively, one can use more sophisticated techniques but at the expense of an increase in the computational cost. We first consider fitting a sigmoid function as an experiment. Then, we consider a nondifferentiable optimization problem.

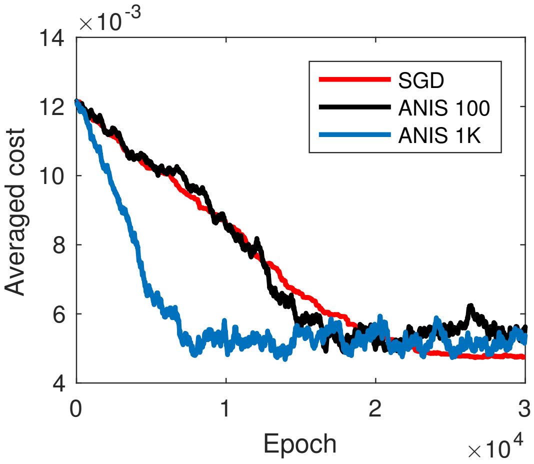

Figure 1: The ANIS algorithm compared to the SGD on a ten-dimensional problem. ANIS-100 denotes ANIS with 100 particles and ANIS 1K is run with $1,000$ particles. Although ANIS-1K seems to converge faster, note that it is much heavier than SGD as it uses 1K evaluations of the cost function at each step. However, SGD needs the gradient information which may not be always available.

4.1 Fitting a sigmoid function

The model used in this experiment is

$$y_k = g_k(\theta) + \epsilon_k = \frac{1}{1 + \exp(\alpha + \beta^\top x_k)} + \epsilon_k$$

where $\epsilon_k \sim \mathcal{N}(0, \gamma^2)$, $x_k \in \mathbb{R}^{d-1}$ denotes the k-th input, and the parameter vector is $\theta = (\alpha, \beta)$ where $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}^{d-1}$. In this experiment, we choose $d = 10$ and set $\gamma^2 = 0.1$ to generate the data from the model. The cost function we attempt to minimize is then

$$f(\theta) = \frac{1}{n} \sum_{k=1}^{n} (y_k - g_k(\theta))^2$$

with $n = 5,000$ where $f_k(\theta) = (y_k - g_k(\theta))^2$. We set $L = 1$, using a single sample at each time for the approximation of $f(\theta)$, which makes algorithms very cheap to run.

We have run two instances of the ANIS method with $N = 100$ and $N = 1,000$ samples, respectively. For both algorithms, we have set $\lambda = 10^{-9}$ and $\sigma^2 = 10^{-2}$. We have used a constant-step-size SGD and tuned the step-size in order to obtain a good performance. Results can be seen in Fig. 1.

As can be seen from the figure, SGD attains a lower error. This phenomenon can be explained with two observations. First, we take the mean as the esti-

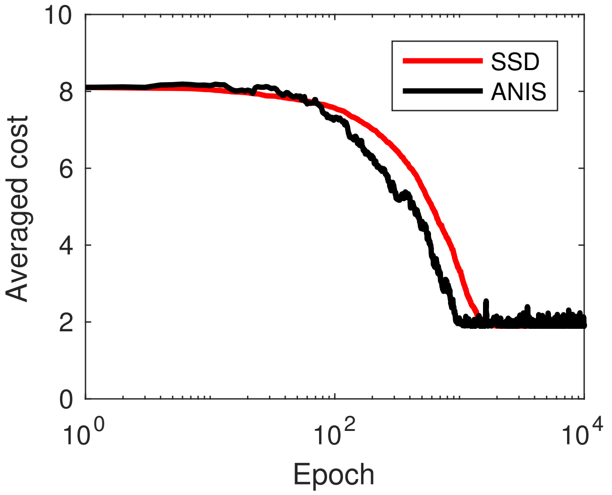

Figure 2: ANIS algorithm on a nondifferentiable 2D problem. We compare it to stochastic subgradient descent (SSD) and conclude that ANIS is able to find good estimates even with 10 samples.

mator, whereas the cost function is not symmetric, so it is not exactly the maximizer of the pdf $\pi(\theta)$. Secondly, as the estimate nears the minimum, it starts fluctuating around the final estimate. One remedy for this would be to decrease the variance of the proposal, in order to focus to the region around minimum, similar to decreasing the step-size of the SGD. However, we do not investigate that option here.

4.2 Large-scale least absolute deviations

In this experiment, we consider a nondifferentiable cost function which comprises many components. The cost function we use in this experiment is

$$f(\theta) = \frac{1}{n} \sum_{k=1}^{n} |y_k - x_k^\top \theta|$$

where $\theta \in \mathbb{R}^2$, $x_k \in \mathbb{R}^2$ and $y_k \in \mathbb{R}$. This problem is also known as *least absolute deviations* (LAD). This cost is not differentiable so gradient-based techniques are not available. However, one can use subgradient-based methods, in the same way as in stochastic gradient techniques [11].

We have run the ANIS scheme with $N = 10$ particles and set $\lambda = 10^{-4}$ and $\sigma^2 = 10^{-4}$. We have chosen a constant step size for the SSD as 5×10^{-3}. The number of data points is $n = 10,000$ for this example. We set $L = 1$, hence sampling a single data point for each iteration. We note that performance can be improved by mini-batching, choosing $L > 1$. The results can be seen from

Fig. 2. For this example, we can conclude that ANIS is a good candidate for nondifferentiable optimization in the large-scale setting. We plan to develop this algorithm in future work.

5 Conclusions

We have proposed an adaptive noisy importance sampler for online optimization problems. The version we have proposed in this paper is the most basic procedure, yet it opens up several interesting future directions. There are many variants of adaptive importance samplers and, using the strategies proposed in the literature, the ANIS algorithm can be improved. Among the candidates, better weight calculations such as [12] or [13] and multiple importance samplers [14] can improve the efficiency of the algorithm drastically. The algorithm can also be applied to distributed settings, where different noisy Monte Carlo estimators can be combined to estimate the minimum [15].

Acknowledgments

This work was partially supported by *Ministerio de Economía y Competitividad* of Spain (TEC2015-69868-C2-1-R ADVENTURE), the Office of Naval Research Global (N62909-15-1-2011), and the regional government of Madrid (program CASICAM-CM S2013/ICE-2845).

References

[1] H. Robbins and S. Monro, “A stochastic approximation method,” *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.

[2] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[3] L. Bottou, F. E. Curtis, and J. Nocedal, “Optimization methods for large-scale machine learning,” *arXiv preprint arXiv:1606.04838*, 2016.

[4] C. P. Robert, *Monte Carlo methods.* Wiley Online Library, 2004.

[5] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi *et al.*, “Optimization by simulated annealing,” *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[6] C. Chen, D. Carlson, Z. Gan, C. Li, and L. Carin, “Bridging the gap between stochastic gradient MCMC and stochastic optimization,” in *Artificial Intelligence and Statistics*, 2016, pp. 1051–1060.

[7] P. Alquier, N. Friel, R. Everitt, and A. Boland, “Noisy Monte Carlo: Convergence of Markov chains with approximate transition kernels,” *Statistics and Computing*, vol. 26, no. 1-2, pp. 29–47, 2016.

[8] F. J. Medina-Aguayo, A. Lee, and G. O. Roberts, “Stability of noisy Metropolis–Hastings,” *Statistics and Computing*, vol. 26, no. 6, pp. 1187–1211, 2016.

[9] M. F. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Miguez, and P. M. Djuric, “Adaptive Importance Sampling: The past, the present, and the future,” *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 60–79, 2017.

[10] O. Cappé, A. Guillin, J.-M. Marin, and C. P. Robert, “Population Monte Carlo,” *Journal of Computational and Graphical Statistics*, vol. 13, no. 4, pp. 907–929, 2004.

[11] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[12] E. Koblents and J. Míguez, “A population Monte Carlo scheme with transformed weights and its application to stochastic kinetic models,” *Statistics and Computing*, vol. 25, no. 2, pp. 407–425, 2015.

[13] V. Elvira, L. Martino, D. Luengo, and M. F. Bugallo, “Improving population Monte Carlo: Alternative weighting and resampling schemes,” *Signal Processing*, vol. 131, pp. 77–91, 2017.

[14] ——, “Generalized multiple importance sampling,” *arXiv preprint arXiv:1511.03095*, 2015.

[15] D. Luengo, L. Martino, V. Elvira, and M. Bugallo, “Efficient linear combination of Partial Monte Carlo estimators,” in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on.* IEEE, 2015, pp. 4100–4104.