On the Application of Danskin's Theorem to Derivative-Free Minimax Problems*

Abdullah Al-Dujaili[1,a)], Shashank Srikant[1,b)], Erik Hemberg[1,c)] and Una-May O'Reilly[1,d)]

[1]*CSAIL, MIT, USA*

[a)]Corresponding author: aldujail@mit.edu
[b)]shash@mit.edu
[c)]hembergerik@csail.mit.edu
[d)]unamay@csail.mit.edu

Abstract. Motivated by Danskin's theorem, gradient-based methods have been applied with empirical success to solve minimax problems that involve non-convex outer minimization and non-concave inner maximization. On the other hand, recent work has demonstrated that Evolution Strategies (ES) algorithms are stochastic gradient approximators that seek robust solutions. In this paper, we address black-box (gradient-free) minimax problems that have long been tackled in a coevolutionary setup. To this end and guaranteed by Danskin's theorem, we employ ES as a stochastic estimator for descent directions. The proposed approach is validated on a collection of black-box minimax problems. Based on our experiments, our method's performance is comparable with its coevolutionary counterparts and favorable for high-dimensional problems. Its efficacy is demonstrated on a real-world application.

INTRODUCTION

Many real-world applications involve an adversary and/or uncertainty, specifically in the security domain. Consequently, several methods have been proposed to find solutions that have the best worst-case (or average) performance for security-critical systems. Important examples include face recognition [1] and malware detection [2]. The notion of security and adversarial robustness can be described by a minimax formulation [3, 4]. The formulation is motivated by theoretical guarantees from Danskin's theorem [5] on using first-order information, i.e. gradients, to find or approximate solutions. Further, where theoretical guarantees can not be assumed, empirical solutions to problems, e.g. digit recognition, have been demonstrated [4].

In this paper, our interest is in black-box (gradient-free) minimax problems where, in contrast to the aforementioned examples of image recognition and malware, gradients are neither symbolically nor numerically available, or they are complex to compute [6]. This has led to extensive use of coevolutionary frameworks [7, 8] to solve such problems. These frameworks however do not reconcile the guarantees provided by gradient-based frameworks in solving the minimax problem. Our goal is to bridge this divide and develop a method for black-box minimax that is consistent with the theoretical assumptions and guarantees of Danskin's theorem while using a gradient estimator in lieu of a gradient. For gradient estimation, we propose to employ a class of black-box optimization algorithms, viz. Evolution Strategies (ES). Our proposition is motivated by the growing body of work [9, 10] which has shown that the performance of gradient-based methods can be rivaled by ES, and that ES is more than just a traditional finite difference approximator [11]. For more empirical and theoretical insights on ES vs. gradient-based methods, see [12, 13].

*The full version of this paper [21] can be found at `https://arxiv.org/pdf/1805.06322.pdf`

FORMAL BACKGROUND

Formally, we are concerned with the black-box minimax optimization problem given a finite budget of function evaluations. Mathematically, the problem is a composition of an inner maximization problem and an outer minimization problem,

$$\min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} \mathcal{L}(\mathbf{x}, \mathbf{y}) \,, \tag{1}$$

where $\mathcal{X} \subseteq \mathbb{R}^{n_x}$, $\mathcal{Y} \subseteq \mathbb{R}^{n_y}$, and $\mathcal{L} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. The problem is called black-box because there is no closed-form expression of $\mathcal{L}$. Instead, one can query an oracle (e.g., a simulation) for the value of $\mathcal{L}$ at a specific pair $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$. The task is then to find the optimal solution $\mathbf{x}^* \in \mathcal{X}$ to Equation 1, whose corresponding objective function value $\mathcal{L}(\mathbf{x}^*, \cdot)$ is at its max at $\mathbf{y}^* \in \mathcal{Y}$, or a good approximate using a finite number of function evaluations, which are expensive in terms of computational resources (e.g. CPU time). The pair $(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \in \mathcal{X} \times \mathcal{Y}$ is called a saddle point of Equation 1 if $\forall \mathbf{x} \in \mathcal{X}$, $\forall \mathbf{y} \in \mathcal{Y}$,

$$\mathcal{L}(\hat{\mathbf{x}}, \mathbf{y}) \leq \mathcal{L}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \leq \mathcal{L}(\mathbf{x}, \hat{\mathbf{y}}) \,. \tag{2}$$

If a saddle point exists, then it follows from Equation 2, that

$$\min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} \mathcal{L}(\mathbf{x}, \mathbf{y}) = max_{\mathbf{y}\in\mathcal{X}} \mathcal{L}(\hat{\mathbf{x}}, \mathbf{y}) = min_{\mathbf{x}\in\mathcal{X}} \mathcal{L}(\mathbf{x}, \hat{\mathbf{y}}) \,. \tag{3}$$

From a game theory perspective, a saddle point represents a two-player zero-sum game equilibrium. Minimax problems with saddle points are referred to as symmetrical problems [14], in contrast to asymmetrical problems for which the condition Equation 2 does not hold. The *regret* of an algorithm's best solution $\mathbf{x}_*$ in comparison to the optimal solution $\mathbf{x}^*$ is defined as

$$r(\mathbf{x}_*) = \max_{\mathbf{y}\in\mathcal{Y}} \mathcal{L}(\mathbf{x}_*, \mathbf{y}) - \mathcal{L}(\mathbf{x}^*, \mathbf{y}^*) \,, \tag{4}$$

where the first term can be computed using an ensemble of off-the-shelf black-box continuous optimization solvers.

METHODS

ES are heuristic search methods inspired by natural evolution. Given a fitness (objective) function, say $f : \mathcal{X} \to \mathbb{R}$, these methods mutate (perturb) a population of genotypes (search points in $\mathcal{X}$) over multiple generations (iterations). At each generation, the fitness of each genotype is evaluated based on f. The fittest genotypes among the current population are then recombined to generate the next population. At the end, the genotype (corresponds to a point in $\mathcal{X}$) with the *best* fitness value is returned as the best point that optimizes f. The notion of "best" refers to the minimum or maximum obtained value of f in a minimization or maximization setup, respectively. Here, we briefly describe one form of ES, in particular a simplified version of natural ES that has recently gained significant attention by the machine learning community [11]. As outlined in Algorithm 3 in the full version of the paper [21], it represents the population with an isotropic Gaussian distribution over the search space $\mathcal{X}$ with mean $\boldsymbol{\mu}$ and *fixed* covariance $\sigma^2 I$ where I is the identity matrix. Over generations, the algorithm aims to maximize the expected fitness value $\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(\boldsymbol{\mu},\sigma^2 I)}[f(\mathbf{x})]$ with respect to the distribution's mean $\boldsymbol{\mu}$ via stochastic gradient ascent using a population size of λ, using the re-parameterization and log-likelihood tricks with $\mathbf{x} = \boldsymbol{\mu} + \sigma\boldsymbol{\epsilon}$ [13, 9]. That is,

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{N}(\boldsymbol{\mu},\sigma^2 I)}[f(\mathbf{x})] = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,I)}[f(\boldsymbol{\mu} + \sigma\boldsymbol{\epsilon})] \tag{5}$$

$$\nabla_{\boldsymbol{\mu}} \mathbb{E}_{\mathbf{x}\sim\mathcal{N}(\boldsymbol{\mu},\sigma^2 I)}[f(\mathbf{x})] = \frac{1}{\sigma} \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,I)}[f(\boldsymbol{\mu} + \sigma\boldsymbol{\epsilon})\boldsymbol{\epsilon}] \tag{6}$$

Descent Direction for Minimax Next, we show that the direction computed by the random perturbations of the current mean can be used to approximate a descent direction of $\max_{\mathbf{y}} \mathcal{L}(\mathbf{x}, \mathbf{y})$. Prior to that and for completeness, we reproduce [4]'s proposition A.2 on the application of Danskin's theorem [5] for minimax problems that are continuously differentiable in $\mathbf{x}$.

Theorem 1 (Madry et al. [4][1])**.** *Let* $\mathbf{y}^* \in \mathcal{Y}$ *and is a maximizer for* $\max_{\mathbf{y}} \mathcal{L}(\mathbf{x}, \mathbf{y})$*. Then, as long as it is nonzero,* $-\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{y}^*)$ *is a descent direction for* $\max_{\mathbf{y}} \mathcal{L}(\mathbf{x}, \mathbf{y})$*.*

[1] Although Theorem 1 assumes $\mathcal{L}$ to be continuously differentiable in $\mathbf{x}$, it has been shown empirically in [4] that breaking this assumption is not an issue in practice.

From Theorem 1 and the assumption that L is twice continuously differentiable in $\mathbf{x}$, we have the following corollary.

Corollary 1. *Let* $\mathbf{y}^* \in \mathcal{Y}$ *and is a maximizer for* $\max_{\mathbf{y}} \mathcal{L}(\mathbf{x}, \mathbf{y})$. *Then, for an arbitrary small* $\sigma > 0$ *and* $\epsilon_1, \ldots, \epsilon_\lambda \sim \mathcal{N}(0, I)$,

$$-\frac{1}{\sigma\lambda} \sum_{i=1}^{\lambda} \mathcal{L}(\mathbf{x} + \sigma\epsilon_i, \mathbf{y}^*)\epsilon_i \tag{7}$$

is a Monte Carlo approximation of a descent direction for $\max_{\mathbf{y}} \mathcal{L}(\mathbf{x}, \mathbf{y})$.

Proof. Without loss of generality, let $\mathcal{X} \subseteq \mathbb{R}$. Then, since σ is arbitrary small, we can approximate $\mathcal{L}$ with a second-order Taylor polynomial,

$$\mathcal{L}(x + \sigma\epsilon, \mathbf{y}^*) \approx \mathcal{L}(x, \mathbf{y}^*) + \mathcal{L}'(x, \mathbf{y}^*)\sigma\epsilon + \frac{1}{2}\mathcal{L}''(x, \mathbf{y}^*)\sigma^2\epsilon^2 \ . \tag{8}$$

Based on Equation 8 and the linearity of expectation, the expectation of $\mathcal{L}(x + \sigma\epsilon, \mathbf{y}^*)\epsilon$ with respect to $\epsilon \sim \mathcal{N}(0, 1)$, $\mathbb{E}_\epsilon[\mathcal{L}(x + \sigma\epsilon, \mathbf{y}^*)\epsilon]$, can be written as

$$\underbrace{\mathbb{E}_\epsilon[\mathcal{L}(x, \mathbf{y}^*)\epsilon]}_{0} + \underbrace{\mathbb{E}_\epsilon[\mathcal{L}'(x, \mathbf{y}^*)\sigma\epsilon^2]}_{\mathcal{L}'(x,\mathbf{y}^*)\sigma} + \frac{1}{2}\underbrace{\mathbb{E}_\epsilon[\mathcal{L}''(x, \mathbf{y}^*)\sigma^2\epsilon^3]}_{0} \ ,$$

where the values of the terms (written under the corresponding braces) come from the definition of *central moments* of the Gaussian distribution [15]. That is, $\mathbb{E}_\epsilon[\mathcal{L}(x + \sigma\epsilon, \mathbf{y}^*)\epsilon] \approx \mathcal{L}'(x, \mathbf{y}^*)\sigma$. Thus, $-\mathcal{L}'(x, \mathbf{y}^*)$, which is—from Theorem 1—a descent direction for $\max_{\mathbf{y}} \mathcal{L}(x, \mathbf{y})$, has a Monte Carlo estimation of the form

$$\begin{aligned} -\mathcal{L}'(x, \mathbf{y}^*) &= -\frac{1}{\sigma}\mathbb{E}_\epsilon[\mathcal{L}(x + \sigma\epsilon, \mathbf{y}^*)\epsilon] \\ &\approx -\frac{1}{\sigma\lambda} \sum_{i=1}^{\lambda} \mathcal{L}(x + \sigma\epsilon_i, \mathbf{y}^*)\epsilon_i \ , \quad \epsilon_i \sim \mathcal{N}(0, 1) \ . \end{aligned}$$

□

Approximating Inner Maximizers While Corollary 1 motivated the use of ES to approximate descent directions for $\max_{\mathbf{y}\in\mathcal{Y}} \mathcal{L}(\mathbf{x}, \mathbf{y})$,[2] an inner maximizer must be computed beforehand. ES can be used for the same. In other words, our use of ES will be of two-fold: 1) computing an inner maximizer $\mathbf{y}^*$ for $\mathcal{L}$; followed by 2) approximating a descent direction on $\mathbf{x}$ for the outer minimization problem along which we proceed to compute the next inner maximizer, and the decent direction therein. However, the inner maximization problem of Equation 1 can be non-concave, for which ES may converge to a local inner maximizer. Restart techniques are common in gradient-free optimization to deal with such cases [16, 17]. Therefore, we use ES with restarts when computing inner maximizers.

Convergence Up to this point, we have seen how ES can be used iteratively to approximate descent directions at inner maximizers of Equation 1. Furthermore, we proposed to address the issue of non-concavity of the inner maximization problem through restarts. One fundamental question is *how much do we step along the descent direction given an inner maximizer?* If we step too much or little, Corollary 1 might not be of help anymore and we may get stuck in cycles similar to the non-convergent behavior of cyclic coordinate descent on certain problems [18]. We investigate this question empirically in our experiments. One should note that cyclic non-convergent behavior is common among coevolutionary algorithms on *asymmetrical* minimax problems [19]. Furthermore, the outer minimization problem can be non-convex. Since we are using ES to approximate the gradient (and eventually the descent direction), we resort to gradient-based restart techniques to deal with cycles and non-convexity of the outer minimization

[2] Current state-of-the-art ES algorithms are far more than stochastic gradient estimators due to their i) ability to automatically adjust the scale on which they sample (step size adaptation) ii) ability to model second order information (covariance matrix adaptation) iii) invariance to rank-preserving transformations of objective values. That is, they do not estimate the gradient, but a related quantity [12]. Our introduction of the simplified version ([21, Algorithm 3]) was to show that a simplified ES algorithm can conform to the guarantees of Theorem 1. In our experiments, we consider some of the established ES variants and compare their efficacy in computing a descent direction for the outer minimization problem.

problem. In particular, we build on Powell's technique [20] to restart whenever the angle between momentum of the outer minimization and the current descent direction is non-positive. One can observe that we employ gradient-*free* restart techniques to solve the inner maximization problem and gradient-*based* counterparts for the outer minimization problem. This is in line with our setup where computing an outer minimizer is guided by the gradient as a descent direction, while approximating the gradient is not a concern for computing an inner maximizer. One should note that search with restarts can still produce local solutions and increasing the number of function evaluations can help in escaping them.

Based on the above, we can now present RECKLESS, our optimization framework for black-box minimax problems. As shown in Algorithm 4 in the full version of this paper [21], the framework comes with 3 parameters: the number of iterations T, the number of function evaluations per iteration v, and the last parameter $s \in (0, 0.5]$ which, along with v, controls how much we should step along the descent direction for the outer minimization problem.

Experiments

To complement its theoretical perspective, a number of numerical experiments were conducted and reported in the full version of this paper [21].

Conclusion

In this paper, we presented RECKLESS: a theoretically-founded framework tailored to black-box problems that are usually solved in a coevolutionary setup. Our proposition employed the stochastic gradient estimation of ES motivated by the experimental success of using gradients at approximate inner maximizers of minimax problems. We found that minimax problems can be solved where outer minimization is given half of the inner maximization's evaluation budget.

REFERENCES

[1] M. Sharif et al., arXiv:1801.00349 (2017).
[2] A. Huang et al., arXiv:1801.02950 (2018).
[3] R. Huang, B. Xu, D. Schuurmans, and C. Szepesvári, arXiv:1511.03034 (2015).
[4] A. Madry et al., "Towards deep learning models resistant to adversarial attacks," in *ICML 2017 Workshop* (2017).
[5] J. M. Danskin, SIAM Journal on Applied Mathematics **14**, 641–664 (1966).
[6] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to derivative-free optimization*, Vol. 8 (Siam, 2009).
[7] J. W. Herrmann, "A genetic algorithm for minimax optimization problems," in *CEC*, Vol. 2 (IEEE, 1999), pp. 1099–1103.
[8] M. T. Jensen, "Robust and flexible scheduling with evolutionary computation," Ph.D. thesis, University of Aarhus 2001.
[9] T. Salimans, J. Ho, X. Chen, and I. Sutskever, arXiv:1703.03864 (2017).
[10] G. Morse et al., "Simple evolutionary optimization can rival stochastic gradient descent in neural networks," in *GECCO* (ACM, 2016), pp. 477–484.
[11] J. Lehman, J. Chen, J. Clune, and K. O. Stanley, arXiv:1712.06568 (2017).
[12] Y. Ollivier, L. Arnold, A. Auger, and N. Hansen, Journal of Machine Learning Research **18**, 1–65 (2017).
[13] D. Wierstra et al., JMLR **15**, 949–980 (2014).
[14] M. T. Jensen, in *Metaheuristics: computer decision-making* (Springer, 2003), pp. 369–384.
[15] A. Winkelbauer, arXiv:1209.4340 (2012).
[16] I. et al. Loshchilov, *International Conference on Parallel Problem Solving from Nature*, 296–305 (2012).
[17] N. Hansen, *GECCO*, 2389–2396 (2009).
[18] M. J. Powell, Mathematical programming **4**, 193–201 (1973).
[19] X. Qiu et al., IEEE Transactions on Cybernetics (2017).
[20] M. J. D. Powell, Mathematical programming **12**, 241–254 (1977).
[21] A. Al-Dujaili et al., `https://arxiv.org/pdf/1805.06322.pdf` ().