ZOERTH-ORDER OPTIMIZATION: COSINE MEASURE IN HIGH DIMENSIONAL SPACE

ZHEKAI LIU

Abstract. Positive spanning is a crucial definition in derivative-free optimization (DFO) problems. A popular way to measure the quality of positive spanning is using the cosine measure [2][4]. However, in reality, how to measure the quality of the different positive spannings in high-dimensional space becomes a challenging nonconvex optimization problem. In this paper, we introduce a novelty **z**ero-**o**rder stochastic algorithm to tackle this **c**osine **m**easure **a**lgorithm problem (**ZO-CMA**). The idea comes from Sign-OPT [3]. The code links are: Zeroth-Order-Cosine-measure-algorithm

1. Introduction

Please see the abstract.

2. Main Problem

Let $\mathbb{D}$ be a non-empty subset of non-zero vectors in $\mathbb{R}^n$. The **cosine measure**[4] of $\mathbb{D}$ is defined as:

$$\mathrm{cm}(\mathbb{D}) = \min_{u\in\mathbb{R}^n, \|u\|=1} \max_{d\in\mathbb{D}} \frac{u^\top d}{\|d\|}.$$

So, giving a positive spanning set $\mathbb{D}$, our problem becomes:

$$\begin{aligned} \min_{u,\alpha} \quad & \alpha \\ \text{s.t.} \quad & \frac{u^\top d_i}{\|d_i\|} \le \alpha, \quad \forall d_i \in \mathbb{D}, \\ & \|u\| = 1. \end{aligned}$$

Here are several methods to solve that in a low-dimensional space.

- Solve this by hand (Solve this non-linear programming problem and also need to enumerate different situations).
- Solve by some package in Python (e.g. CVXPY).

The advantage of the first method is that we could avoid the local minimum in lower dimensions, so it is a promising method to tackle a low-dimensional space problem. For instance, your question:

$$\mathbb{D} = \{e_1, e_2, e_1 + e_2 + e_3, e_4, -e_1 - e_2, -e_3 - e_4\},$$

however, because we need to consider and enumerate all situations during this non-linear programming problem, things will get harder in high-dimensional space.

Furthermore, because of the nonconvex property of the objective function in high-dimensional space, CVXPY(usually used to solve convex problems) isn't a good choice.

Date: Feb. 2025.

3. High-dimensional Space Situation

3.1. **Methodology.** Give this optimization problem below:

$$\min_{u\in\mathbb{R}^n,\|u\|=1}\max_{d\in\mathbb{D}}\frac{u^\top d}{\|d\|},$$

where $\mathbb{D}$ is a positive spanning, for abbreviation, we denote our object function as $f(u)$. This optimization problem is a Min-Max problem, which means that we could not use the traditional optimization method (white-box gradient-based optimization) to solve it.

Building on these insights, several popular approaches have emerged [5],[1],[3]. In this paper, we used the Zoerth-order gradient estimation in the [5]:

$$\hat{g}_u = \frac{f(u+\mu\mathbf{u}) - f(u-\mu\mathbf{u})}{2\mu}\mathbf{u},$$

where $\mu > 0$ is the smoothing parameter, which is a small positive constant, and $\mathbf{u} \in \mathbb{R}^d$ is distributed in $\mathcal{N}(0, \mathbf{I}_d)$.

Actually, this is a really common problem in machine learning, and the problem is called adversarial attack. In computer vision's perspective, a vector u is just like an image with a noise. Our goal is to let the pre-trained neural network or convolutional network model have a wrong classification of this image [3].

Algorithm 1 ZO-CMA: Rectangular Coordinates Vision

Input: Positive spanning set $\mathbb{D}$, Q, I, T are two constants, μ is also a constant in Zoerth-order gradient, learning rate η, and the object function f.

Initialization:

1: Randomly sample $r_1, \cdots, r_Q$ from a Uniform distribution on unit shpere $\mathcal{S}(0,1)$.

2: $u_0 = \arg\min_{\{r_1,\cdots,r_Q\}} f(r_i);\ i = 1, \cdots, Q$.

Optimization:

3: **for** t in 0 to $T-1$ **do**

4: **for** i in 1 to I **do**

5: $g_i^t = \dfrac{f(u_t+\mu\mathbf{u}_i) - f(u_t-\mu\mathbf{u}_i)}{2\mu}\mathbf{u}_i$, where $\mathbf{u}_i \sim \mathcal{S}(0,1)$.

6: $g_u^t = \dfrac{1}{I}\sum_{i=1}^{I} g_i^t$.

7: **end for**

8: $\hat{u}^{t+1} = u^t - \eta g_u^t$.

9: $u^{t+1} = \dfrac{\hat{u}^{t+1}}{\|u^{t+1}\|}$

10: **end for**

11: **return** u^T

Explanation: This version of the algorithm optimizes the coordinate position of vector u directly. We normalize the vector in step 9 for each iteration to ensure that the vector is always a unit vector. The reason we do not use the normal distribution in step 5 is that we aim to reduce the oscillation.

Algorithm 2 ZO-CMA: Spherical Coordinates Vision

Input: Positive spanning set $\mathbb{D}$, Q, I, T are two constants, μ is also a constant in Zoerth-order gradient, learning rate η, and the object function f.

Initialization:

Randomly sample $r_1, \cdots, r_Q$ from a Uniform distribution between $(-\pi, \pi]$.

$\theta_0 = \arg\min_{\{r_1, \cdots, r_Q\}} f(r_i);\ i = 1, \cdots, Q.$

Optimization:

for t in 0 to $T-1$ **do**

  for i in 1 to I **do**

    $g_i^t = \dfrac{f(\theta_t + \mu \mathbf{u}_i) - f(\theta_t - \mu \mathbf{u}_i)}{2\mu} \mathbf{u}_i$, where $\mathbf{u}_i \sim \mathcal{S}(0, 1)$.

    $g_\theta^t = \dfrac{1}{I} \sum_{i=1}^{I} g_i^t$.

  end for

  $\theta^{t+1} = \theta^t - \eta g_\theta^t$.

end for

return θ^T

Clarification: The ZO-CMA spherical coordinates Vision is more reasonable because: 1) Our optimized field is on a unit sphere instead of an unconstrained plane, so optimizing the angle of the vector u in spherical coordinates is a promising option. 2) We can guarantee that the vector u is on the unit sphere surface for each iteration rather than normalize it for each iteration, so **Algorithm 2** is more efficient compared with **Algorithm 1**. The idea of **Algorithm** 2 comes from adversarial attack Sign-OPT [3] directly.

3.2. **Convergence Analysis.**

Theorem 1. *Under the local gradient's Lipschitz assumption* $f \in l^{1+}, x \in B_\Delta$, *let learning rate* $\eta = \dfrac{1}{L^2}$, *when* $T = \mathcal{O}(\dfrac{2L^2}{\varepsilon^2})$, $\mu = \dfrac{\varepsilon}{\sqrt{DLd}}, I = \mathcal{O}(\dfrac{\sigma^2}{2L^2\varepsilon^2}), \mu = \mathcal{O}(\sqrt{\dfrac{\varepsilon}{Ld}})$, *where* L *represents Lipchitz constant and* d *represents the dimensions, then we have:*

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \|\nabla f(u^t)\|^2 \leq \mathcal{O}(\varepsilon^2)$$

The vector u *will converge to the critical position (this point will change* $d \in \mathbb{D}$ *to measure cosine measure), so the local gradient Lipschitz is reasonable to this problem.*

Proof. Please see the appendix. □

4. Experiments

Clearification: Actually, I'm not quite sure who also did this problem in the numerical aspect, so there is no baseline to compare.

Problem 1: $\mathbb{D} = \{e_1, e_2\}$, it's easy to calculate that the cosine measure of this positive spanning is:

$$f(u) = cos(135^\circ) \approx= \textbf{-0.707}$$

By using **ZO-CMA** we have following result after :

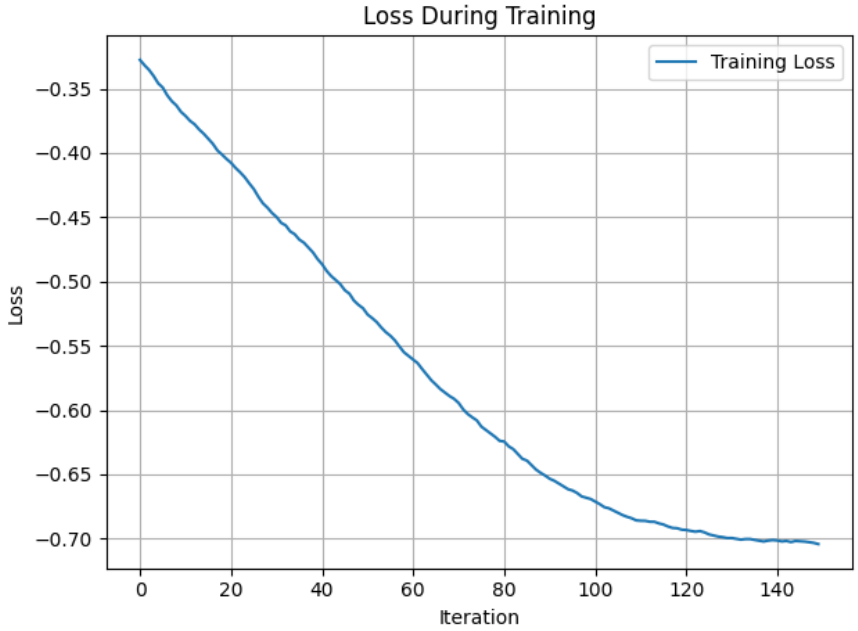

FIGURE 1. ZO-CMA Recrangular Vision: $I = 10, Q = 2, T = 150, \mu = 0.1, \eta = 0.01$

In this experiment, we only use 150 iterations with $3,002$ function evaluations($3,000+2$, 2 for initialization). After 150 iterations, $f(u) \approx$ **-0.704**.

Problem 2: $\mathbb{D} = \{e_1, e_2, e_1+e_2+e_3, e_4, -e_1-e_2, -e_3-e_4\}$. Firstly, we random sample vector on the unit sphere for $10,000$ times, and the final result is $f(u) \approx$ **0.1745**.(I need to apologize that I didn't calculate it by hand.)

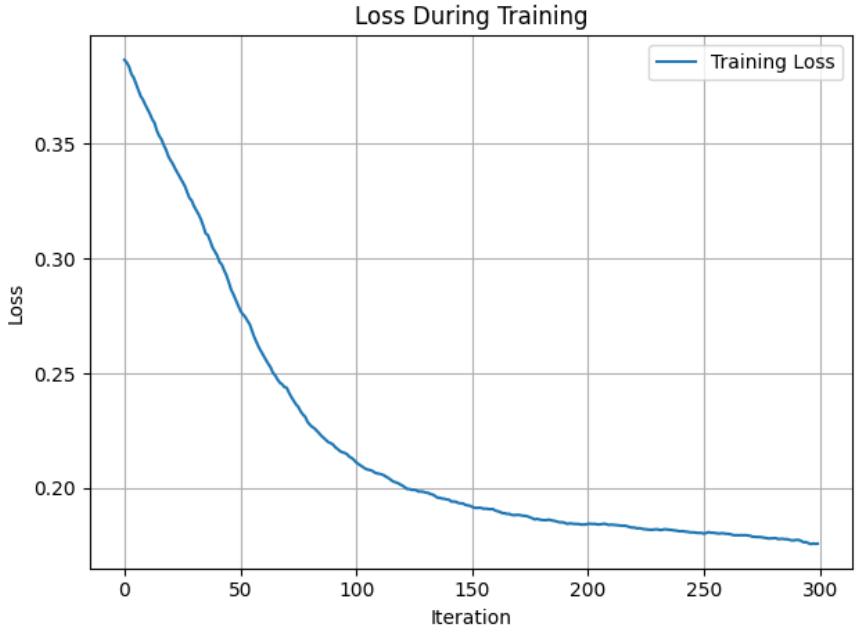

FIGURE 2. ZO-CMA Rectangular Vision: $I = 10, Q = 30, T = 300, \mu = 0.1, \eta = 0.01$

In this experiment, we use 300 iterations with $6,030$ function evaluations$(6,000+30$, $6,030$ for initialization). After 300 iterations, $f(u) \approx$ **0.1759**.

Problem 3: $\mathbb{D} = \{e_i; i = 1, 2, \cdots, 8\}$. Firstly, we random sample vector on the unit sphere for **1,000,000** times, and the final result is $f(u) \approx$ **-0.3084**. Actually, this is NOT the correct anwer, because there are enormous situations. So, here we use another method:

Step 1: Maximization For any unit vector $u = (u_1, u_2, \ldots, u_8)$, since each $d \in \mathbb{D}$ is a standard basis vector, we obtain:

$$\max_{d\in\mathbb{D}} \frac{u^\top d}{\|d\|} = \max_{i=1,\ldots,8} u_i.$$

Step 2: Minimization Now, we solve:

$$\min_{\|u\|=1} \max_{i=1,\ldots,8} u_i.$$

To minimize the maximum component of u, an optimal choice is to distribute the components of u evenly with equal negative values:

$$u = \left(-\frac{1}{\sqrt{8}}, -\frac{1}{\sqrt{8}}, \ldots, -\frac{1}{\sqrt{8}}\right).$$

Therefore, the cosine measure of $\mathbb{D}$ is:

$$\mathrm{cm}(\mathbb{D}) = -\frac{1}{\sqrt{8}} \approx \textbf{-0.3536}.$$

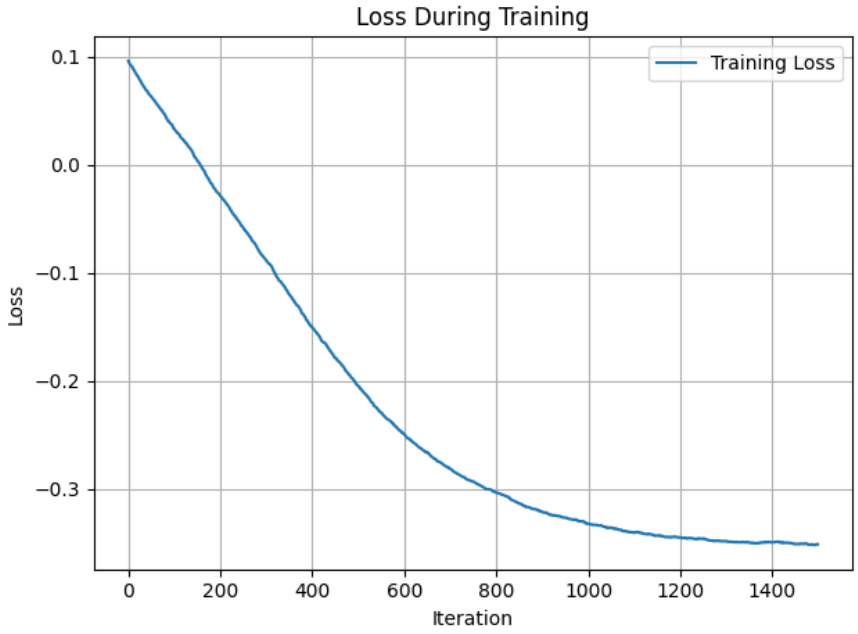

FIGURE 3. ZO-CMA Rectangular Vision: $I = 10, Q = 30, T = 1,500, \mu = 0.1, \eta = 0.01$

In our experiment, we use $1,500$ iterations with $30,030$ function evaluations$(30,000+$ 30, 30 for initialization). After $1,500$ iterations, $f(u) \approx$ **-0.3511**.

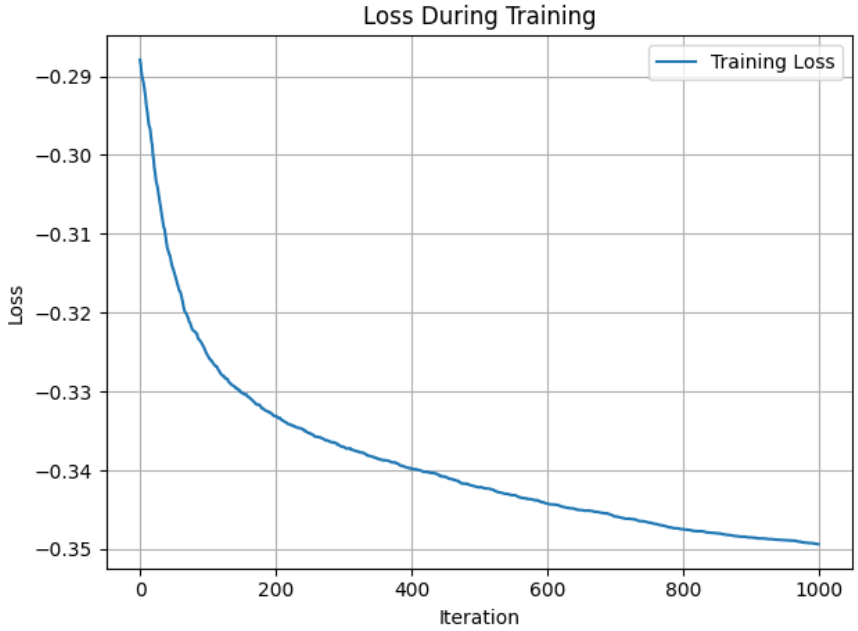

FIGURE 4. ZO-CMA Spheric Vision: $I = 10, Q = 30, T = 1,000, \mu = 0.1, \eta = 0.01$

In spheric coordinates vision experiments (Algorithm 2), we use $1,000$ iterations with $21,000$ function evaluations($20,000 + 1,000$, $1,000$ for initialization). After $1,000$ iterations, $f(u) \approx$ **-0.3494**.

In these experiments, we found that initialization is a crucial part of the algorithm, a relatively large number of initialization could reduce the possibility of getting stuck in the local minimum.

Furthermore, I rarely tuned the hyperparameter in these experiments, actually, the function evaluation times could dramatically decrease by raising the learning rate η, however raising the algorithm's variance instead [6].

5. Limitation

The **ZO-CMA** could only find the local minimum solution in reality. During the experiments, we need to reimplement the codes several times to achieve better performance. Furthermore, our algorithm is almost a monotonic decrease but may increase at some spots.

6. Conclusion

In this work, we introduce a novelty **a**lgorithm framework by using the **z**eroth-**o**rder gradient to estimate the **c**osine **m**easure (ZO-CMA). We used both experiments and theory to prove the effectiveness of our method.

References

[1] Charles Audet and Warren Hare. *Derivative-Free and Blackbox Optimization*. Cham, Switzerland: Springer, 2017. DOI: 10.1007/978-3-319-68913-5. URL: https://dx.doi.org/10.1007/978-3-319-68913-5.

[2] Charles Audet, Warren Hare, and Gabriel Jarry-Bolduc. "The cosine measure relative to a subspace". In: *arXiv preprint arXiv:2401.09609* (2024).

[3] Minhao Cheng et al. "Sign-opt: A query-efficient hard-label adversarial attack". In: *arXiv preprint arXiv:1909.10773* (2019).
[4] Tamara G. Kolda, Robert Michael Lewis, and Virginia Torczon. "Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods". In: *SIAM Review* 45.3 (2003), pp. 385–482. DOI: 10.1137/S003614450242889.
[5] Yurii Nesterov and Vladimir Spokoiny. "Random gradient-free minimization of convex functions". In: *Foundations of Computational Mathematics* 17.2 (2017), pp. 527–566.
[6] D.H. Wolpert and W.G. Macready. "No free lunch theorems for optimization". In: *IEEE Transactions on Evolutionary Computation* 1.1 (1997), pp. 67–82. DOI: 10.1109/4235.585893.

Appendix A. Appendix

Assumption 1 [L-Smoothness] For convenience, denote:

$$f(u) := \min_{u\in\mathbb{R}^n,\|u\|=1} \max_{\mathbf{d}\in\mathbb{D}} \frac{u^\top \mathbf{d}}{\|\mathbf{d}\|}.$$

Because the object function is $\min\max cos(\cdot,\cdot)$, it is reasonable to assume that the object function f is a local Lipchitz function.

$$\|\nabla f(u) - \nabla f(u')\| \le L\|u - u'\|,\ u, u' \in B_\Delta. \tag{1}$$

Where, $u \in \mathbb{R}^d$, L represents the Lipchitz constant, and B_Δ is a trust region. This condition is equivalent to the following inequality:

$$|f(u') - f(u) - \langle \nabla f(u), u' - u\rangle| \le \frac{L^2}{2}\|u' - u\|^2, \quad u, u' \in B_\Delta. \tag{2}$$

Lemma 1. *(Theorem 1 and Lemma 3 in [5]) If $f \in C^{1,1}(\mathbb{R}^d)$ with constant L, and $x \in \mathbb{R}^d$ then for the bias with respect to the zeroth-order function:*

$$|f_\mu(x) - f(x)| \le \frac{\mu^2}{2} Ld. \tag{3}$$

For the bias with respect to the zeroth-order gradient:

$$\|\nabla f_\mu(x) - \nabla f(x)\| \le \frac{\mu}{2} L(d+3)^{3/2}, \tag{4}$$

where:

$$f_\mu(x) = \frac{1}{\kappa}\int_{\mathbb{R}^d} f(x + \mu u) e^{-\frac{1}{2}\|u\|^2} \mathrm{d}u;$$

$$\nabla f_\mu(x) = \frac{1}{\kappa}\int_{\mathbb{R}^d} \frac{f(x+\mu u) - f(x-\mu u)}{2\mu} e^{-\frac{1}{2}\|u\|^2} u \mathrm{d}u.$$

According to Eq.(12) in [5], we can now that f_μ is Lipschitz smooth:

$$\|\nabla f_\mu(u) - \nabla f_\mu(u')\| \le L\|u - u'\|, \tag{5}$$

where $\nabla f_\mu(u) = \frac{1}{\kappa}\int_{\mathbb{R}^d} \frac{f(x+\mu u) - f(x-\mu u)}{2\mu} e^{-\frac{1}{2}\|\mathbf{u}\|^2} u \mathrm{d}u$, *and* $\kappa = \int_{\mathbb{R}^d} e^{-\frac{1}{2}\|u\|^2} \mathrm{d}u$.

Convergence analysis

Smoothness: $|f(u') - f(u) - \langle \nabla f(u), u' - u\rangle| \le \frac{L^2}{2}\|u' - u\|^2, \quad u, u' \in B_\Delta$. Firstly, because of the smoothness of the $f(\cdot)$ with respect to u from the inequality (2) and the

bias from the zeroth-order gradient from the inequality (6):

$$\begin{aligned} f(u^{t+1}) - f(u^t) \leq & \left(f_\mu(u^{t+1}) + \frac{\mu^2}{2} Ld \right) - \left(f_\mu(u^t) - \frac{\mu^2}{2} Ld \right) \\ = & \ f_\mu(u^{t+1}) - f_\mu(u^t) + \mu^2 Ld \\ \leq & \langle \nabla f_\mu(u^t), u^{t+1} - u^t \rangle + \frac{L^2}{2} \|u^{t+1} - u^t\|^2 + \mu^2 Ld. \end{aligned}$$

Let $\eta = \frac{1}{L^2}$, denote the variance $\frac{\sigma^2}{I} = \mathbb{E}_u \|\hat{g}_\mu(u) - \nabla f_\mu(u^t)\|^2$,

$$\begin{aligned} \mathbb{E} f(u^{t+1}) - f(u^t) \leq & -\frac{1}{L^2} \mathbb{E} \|\hat{g}_\mu(u^t)\|^2 + \frac{1}{2L^2} (\mathbb{E} \|\hat{g}_\mu(u^t) - \nabla f_\mu(u^t)\|^2 + \mathbb{E} \|\nabla f_\mu(u^t)\|^2) + \mu^2 Ld \\ \leq & -\frac{1}{2L^2} \|\nabla f(u^t)\|^2 + \frac{\sigma^2}{2L^2 I} + 2\mu^4 L^2 d^2 \end{aligned}$$

At this moment denote $C = 2\mu^4 L^2 (d)^2$:

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \|\nabla f(u^t)\|^2 \leq & \ 2L^2 \frac{1}{T} \sum_{t=0}^{T-1} (f(u^0) - f^*) + C \\ = & \ \mathcal{O}(\varepsilon^2) \end{aligned}$$

When $T = \mathcal{O}(\frac{2L^2}{\varepsilon^2})$, $\mu = \frac{\varepsilon}{\sqrt{DLd}}, I = \mathcal{O}(\frac{\sigma^2}{2L^2\varepsilon^2}), \mu = \mathcal{O}(\sqrt{\frac{\varepsilon}{Ld}})$, and f^* represents minimum valueof the object function $f(u)$. □