

# Covariance Matrix Adaptation - Evolution Strategy:

## A summary

Gaëtan Serré

ENS Paris-Saclay - Centre Borelli

gaetan.serre@ens-paris-saclay.fr

### 1. Introduction

The *Covariance Matrix Adaptation - Evolution Strategy* (CMA-ES) is a global and black-box optimization algorithm. It is a randomized black-box search algorithm as it samples evaluation points from a distribution conditioned with the previous parameters. This kind of algorithm is detailed in Algorithm 1. CMA-ES uses a multivariate Gaussian as the sampling distribution  $\mathcal{N}(m, C)$ . The authors made this choice as, given the variances and covariances between components, the normal distribution has the largest entropy in  $\mathbb{R}^d$ . To goal is to find how update the mean and covariance matrix of this distribution to minimize the trade-off between finding a good approximation of the optimum and evaluate the objective function as few times as possible. In this small paper, we will present the ideas behind CMA-ES. For more details, see (Hansen 2023). Throughout this paper, we will suppose that the objective function is to be maximized.

For  $g$  in  $1 \dots k$ :

1. Let  $d_\theta$  a distribution on  $\mathcal{X}$  parametrized by  $\theta$ ;
2. Sample  $\lambda$  points:  $(x_i)_{1 \leq i \leq \lambda} \sim d_{\theta_i}$ ;
3. Evaluate the points:  $f((x_i)_{1 \leq i \leq \lambda})$ ;
4. Update the parameters  $\theta_{i+1} = F(\theta_i, (x_1, f(x_1)), \dots, (x_\lambda, f(x_\lambda)))$ .

Algorithm 1: Black-box search algorithm.

### 2. Update the mean

In the CMA Evolution Strategy, the  $\lambda$  points are sampled from a multivariate Gaussian distribution which writes:

$$(x_i^{(g)})_{1 \leq i \leq \lambda} \sim m^{(g)} + \sigma^{(g)} \mathcal{N}(0, C^{(g)}), \quad (1)$$

where  $g$  is the generation number,  $m^{(g)}$  is the mean vector,  $\sigma^{(g)}$  is the “overall” standard deviation and  $C^{(g)}$  is the covariance matrix. It is equivalent to say that  $x_i^{(g)} \sim \mathcal{N}(m^{(g)}, (\sigma^{(g)})^2 C^{(g)})$ .

To update the mean, we begin by selecting the  $\mu$  best points, i.e.:

$$f(x_1^{(g)}) \geq \dots \geq f(x_\mu^{(g)}) \geq f(x_{\mu+1}^{(g)}) \geq \dots f(x_\lambda^{(g)}). \quad (2)$$

We introduce the index notation  $i : \lambda$ , denoting the index of the  $i$ -th best point. The mean at generation  $g + 1$  becomes a weighted average of those points:

$$m^{(g+1)} = m^{(g)} + c_m \sum_{i=1}^{\mu} w_i (x_{i:\lambda} - m^{(g)}), \quad (3)$$

where:

$$\sum_{i=1}^{\mu} w_i = 1, \quad w_1 \geq \dots \geq w_{\mu} \geq 0, \quad (4)$$

and  $c_m$  is a learning rate, usually set to 1. In that case, Eq. 3 simply becomes:

$$m^{(g+1)} = \sum_{i=1}^{\mu} w_i x_{i:\lambda}. \quad (5)$$

The choice of the weights is crucial in CMA-ES as they represent the trade-off between exploration and exploitation. To do so, we define the quantity  $\mu_{\text{eff}}$  as:

$$\mu_{\text{eff}} = \left( \frac{\|w\|_1}{\|w\|_2} \right)^2 = \frac{\left( \sum_{i=1}^{\mu} |w_i| \right)^2}{\sum_{i=1}^{\mu} w_i^2} = \frac{1}{\sum_{i=1}^{\mu} \frac{w_i^2}{w_i}}. \quad (6)$$

From Eq. 4, one can easily derive  $1 \leq \mu_{\text{eff}} \leq \mu$ , the latter happens when all the weights are equal, i.e.  $\forall 1 \leq i \leq \mu, w_i = \frac{1}{\mu}$ .  $\mu_{\text{eff}}$  quantize the loss of variance due to the selection of the best points. According to the author,  $\mu_{\text{eff}} \approx \frac{\lambda}{4}$  indicates a reasonable choice of  $w_i$ . A simple and decent way to achieve that is to set  $w_i \propto \mu - i + 1$  (see 4.1). Choosing  $c_m < 1$  can work well on noisy function. However, the step-size  $\sigma$  is roughly proportional to  $\frac{1}{c_m}$  and thus, with a too small  $c_m$ , the search would moves away from the current region of relevance.

### 3. Update the covariance matrix

To update the covariance matrix, we need to estimate it using the points  $(x_i)_{1 \leq i \leq \lambda}$ . In this section, we assume  $\sigma = 1$  for simplicity. If  $\sigma \neq 1$ , one can simply rescale the covariance matrix by  $\frac{1}{\sigma^2}$ . If we have enough sample, one can use the empirical covariance matrix:

$$C_{\text{emp}}^{(g+1)} = \frac{1}{\lambda - 1} \sum_{i=1}^{\lambda} \left( x_i^{(g+1)} - \frac{1}{\lambda} \sum_{i=1}^{\lambda} x_i^{(g+1)} \right) \left( x_i^{(g+1)} - \frac{1}{\lambda} \sum_{i=1}^{\lambda} x_i^{(g+1)} \right)^{\top}. \quad (7)$$

A different would be to use the real mean  $m^{(g+1)}$  computed before instead of the empirical mean:

$$C_{\lambda}^{(g+1)} = \frac{1}{\lambda} \sum_{i=1}^{\lambda} \left( x_i^{(g+1)} - m^{(g+1)} \right) \left( x_i^{(g+1)} - m^{(g+1)} \right)^{\top}. \quad (8)$$

Both are unbiased estimators of the covariance matrix. However, they do not influence the search towards the direction of the  $\mu$  best points. To do so, one can use the same *weighted selection* as in Eq. 3 :

$$C_{\mu}^{(g+1)} = \sum_{i=1}^{\mu} w_i \left( x_{i:\lambda}^{(g+1)} - m^{(g)} \right) \left( x_{i:\lambda}^{(g+1)} - m^{(g)} \right)^{\top}. \quad (9)$$

This last estimator tends to reproduce the current best points and thus allows a faster convergence. However, this estimation method requires a lot of samples and  $\mu_{\text{eff}}$  must be large enough to be reliable. The author suggests another method to estimate  $C^{(g+1)}$  that tackles these two issues, the *rank- $\mu$*  method.

#### 3.1. Rank- $\mu$ method

As stated before, for Eq. 9 to be a reliable estimator, one need a lot of sample, as ideally  $\mu_{\text{eff}} \approx \frac{\lambda}{4}$ . However, evaluate the function is often the main bottleneck of the algorithm. The author suggests

to use the *rank- $\mu$*  method to estimate the covariance matrix. The idea behind this method is to use information of previous generations in the estimation of the next one:

$$C^{(g+1)} = \frac{1}{g+1} \sum_{i=0}^g \frac{1}{\sigma^{(i)^2}} C_\mu^{(i+1)}, \quad (10)$$

where  $\sigma^{(i)}$  is the step-size at generation  $i$ . In Eq. 10, each generation has the same weight in the estimation of the covariance matrix of the next generation. A natural idea would be to give more weight to the most recent generations through exponential smoothing:

$$C^{(g+1)} = (1 - c_\mu) C^{(g)} + c_\mu \frac{1}{\sigma^{(g)^2}} C_\mu^{(g+1)}, \quad (11)$$

where  $c_\mu \leq 1$  is a learning rate. The author suggests that  $c_\mu \approx \min\left(1, \frac{\mu_{\text{eff}}}{d^2}\right)$  is a reasonable choice. Eq. 11 can be written as:

$$C^{(g+1)} = (1 - c_\mu) C^{(g)} + c_\mu \sum_{i=1}^{\mu} w_i y_{i:\lambda}^{(g+1)} y_{i:\lambda}^{(g+1)\top}, \quad (12)$$

where  $y_{i:\lambda}^{(g+1)} = \frac{x_{i:\lambda}^{(g+1)} - m^{(g)}}{\sigma^{(g)}}$ . This method is so called *rank- $\mu$*  as the rank of the sum of the dot products is  $\min(\mu, d)$ . Finally, the author generalizes Eq. 12 with  $\lambda$  weights that does not requires to sum to 1 nor being positive:

$$C^{(g+1)} = \left(1 - \sum_{i=1}^{\lambda} w_i c_\mu\right) C^{(g)} + c_\mu \sum_{i=1}^{\lambda} w_i y_{i:\lambda}^{(g+1)} y_{i:\lambda}^{(g+1)\top}. \quad (13)$$

Usually,  $\sum_{i=1}^{\mu} w_i = 1 = -\sum_{i=\mu+1}^{\lambda} w_i$ . To emphasize the importance of  $c_\mu$ , the author introduce the *backward time horizon*  $\Delta g$ . It represent the number of generations used to encode roughly 63% of the information of the estimation of the covariance matrix of the next generation. E.g. if  $\Delta g = 10$ , it means that the 10 last generations are used to compute 63% of the information of the covariance matrix of the next generation. Indeed, Eq. 11 can be extended to:

$$C^{(g+1)} = (1 - c_\mu)^{(g+1)} C^{(0)} + c_\mu \sum_{i=0}^g (1 - c_\mu)^{g-i} \frac{1}{\sigma^{(i)^2}} C_\mu^{(i+1)}. \quad (14)$$

Therefore,  $\Delta g$  is defined by:

$$c_\mu \sum_{i=g+1-\Delta g}^g (1 - c_\mu) \approx 0.63 \approx 1 - \frac{1}{e}. \quad (15)$$

One can solve this equation to find  $\Delta g \approx \frac{1}{c_\mu}$  (see 4.2). It shows that, the smaller is  $c_\mu$  the more past generations are used to compute the covariance matrix.

## Bibliography

N. Hansen, “The CMA Evolution Strategy: A Tutorial,” arXiv, 2023. Accessed: Apr. 4, 2023. [Online]. Available: <http://arxiv.org/abs/1604.00772> (Comment: ArXiv e-prints, arXiv:1604.00772, 2016, pp.1-39)

## 4. Annex

### 4.1. $\mu_{\text{eff}}$

*Proof.* Let  $w_i = \frac{\mu-i+1}{\sum_{i=1}^{\mu} \mu-i+1}$ . Then:

$$\begin{aligned}
 \mu_{\text{eff}} &= \frac{1}{\sum_{i=1}^{\mu} \left( \frac{\mu-i+1}{\sum_{i=1}^{\mu} \mu-i+1} \right)^2} = \frac{1}{\sum_{i=1}^{\mu} \left( \frac{i}{\sum_{i=1}^{\mu} i} \right)^2} \\
 &= \frac{1}{\sum_{i=1}^{\mu} \frac{i^2}{\left( \frac{\mu(\mu+1)}{2} \right)^2}} = \frac{1}{\frac{\mu(\mu+1)(2\mu+1)}{6 \left( \frac{\mu(\mu+1)}{2} \right)^2}} \\
 &= \frac{6 \left( \frac{\mu(\mu+1)}{2} \right)^2}{\mu(\mu+1)(2\mu+1)} = \frac{3\mu(\mu^3 + 2\mu^2 + \mu)}{2(2\mu^3 + 3\mu^2 + \mu)} \\
 &= \frac{3\mu(1 + \mu)}{2(1 + 2\mu)} \approx \frac{3\frac{\lambda}{2}(1 + \frac{\lambda}{2})}{2(1 + \lambda)} \\
 &= \frac{\frac{3\lambda^2 + 6\lambda}{2}}{4(1 + \lambda)} = \frac{3\lambda(2 + \lambda)}{8(1 + \lambda)} \\
 &\approx \frac{3\lambda}{8}.
 \end{aligned} \tag{16}$$

■

### 4.2. $\Delta g$

*Proof.*

$$\begin{aligned}
 c_{\mu} \sum_{i=g+1-\Delta g}^g (1 - c_{\mu})^{g-i} &= c_{\mu} \left( (1 - c_{\mu})^{\Delta g-1} + (1 - c_{\mu})^{\Delta g-2} + \dots + (1 - c_{\mu})^0 \right) \\
 &= c_{\mu} \sum_{i=0}^{\Delta g-1} (1 - c_{\mu})^i \\
 &= c_{\mu} \frac{(1 - c_{\mu})^0 - (1 - c_{\mu})^{\Delta g}}{1 - (1 - c_{\mu})} \\
 &= c_{\mu} \frac{1 - (1 - c_{\mu})^{\Delta g}}{c_{\mu}} = 1 - (1 - c_{\mu})^{\Delta g}.
 \end{aligned} \tag{17}$$

Thus, the problem becomes to find  $\Delta g$  such that  $1 - (1 - c_{\mu})^{\Delta g} \approx 0.63 \approx 1 - \frac{1}{e}$ :

$$\begin{aligned}
1 - (1 - c_\mu)^{\Delta g} &= 1 - \frac{1}{e} \\
\Leftrightarrow (1 - c_\mu)^{\Delta g} &= e^{-1} \\
\Leftrightarrow \Delta g \ln(1 - c_\mu) &= -1 \\
\Leftrightarrow \Delta g &= -\frac{1}{\ln(1 - c_\mu)} \approx \frac{1}{c_\mu} \text{ (using Taylor's expansion of order 1).}
\end{aligned} \tag{18}$$

■