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 conditionned with the previous parameters. This kind of algorithm is detailed in Figure 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 $1 \ldots k$:

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

Algorithm 1: Black-box search algorithm.

2. Update the mean

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

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

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 $x_i^{(g)} \sim \mathcal{N}\left(m^{(g)}, \left(\sigma^{(g)}\right)^2 C^{(g)}\right)$.

To update the mean, we begin by selecting the μ best points, i.e. :

$$f\left(x_1^{(g)}\right) \geq \ldots \geq f\left(x_\mu^{(g)}\right) \geq f\left(x_{\mu+1}^{(g)}\right) \geq \ldots f\left(x_\lambda^{(g)}\right). \tag{2}$$

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:

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

where:

$$\sum_{i=1}^{\mu} w_i = 1, \quad w_1 \geq ... \geq w_\mu \geq 0, \tag{4}$$

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

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

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 μ_{eff} as:

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

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. $\forall 1 \leq i \leq \mu, w_i = \frac{1}{\mu}$. μ_{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 Eq. 10). Choosing $c_m < 1$ can work well on noisy function. However, the step-size σ 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:

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

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

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

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

$$C_{\mu}^{(g+1)} = \sum_{i=1}^{\mu} w_i \left(x_i^{(g+1)} - m^{(g+1)}\right) \left(x_i^{(g+1)} - m^{(g+1)}\right)^\top. \tag{9}$$

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 μ_{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-μ* method.

3.1. Rank-μ method

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

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

$$\begin{aligned}
\mu_{\text{eff}} &= \frac{1}{\sum_{i=1}^{\mu} \left(\frac{\mu - i + 1}{\sum_{i=1}^{\mu} \mu - i + 1} \right)^2} = \frac{1}{\sum_{i=1}^{\mu} \left(\frac{i}{\sum_{i=1}^{\mu} i} \right)^2} \\
&= \frac{1}{\sum_{i=1}^{\mu} \frac{i^2}{\left(\frac{\mu(\mu+1)}{2} \right)^2}} = \frac{1}{\frac{\mu(\mu+1)(2\mu+1)}{6\left(\frac{\mu(\mu+1)}{2} \right)^2}} \\
&= \frac{6\left(\frac{\mu(\mu+1)}{2} \right)^2}{\mu(\mu+1)(2\mu+1)} = \frac{3\mu(\mu^3 + 2\mu^2 + \mu)}{2(2\mu^3 + 3\mu^2 + \mu)} \\
&= \frac{3\mu(1+\mu)}{2(1+2\mu)} \approx \frac{3\frac{\lambda}{2}\left(1 + \frac{\lambda}{2}\right)}{2(1+\lambda)} \\
&= \frac{\frac{3\lambda^2 + 6\lambda}{2}}{4(1+\lambda)} = \frac{3\lambda(2+\lambda)}{8(1+\lambda)} \\
&\approx \frac{3\lambda}{8}.
\end{aligned} \tag{10}$$