The Covariance Matrix Adaptation Evoluion Strategy

Authors:
Villads Rendtorff Zimakoff
Daniel Holm Johansen
Silas Larsen Kapetanopoulos

Advanced Methods in Applied Statistics

March 5, 2025

Introduction

A powerful tool in statistics is the Maximum Likelihood Estimator (MLE) which is great for estimating the function parameters given the likelihood for a series of data points x_i. So far we primarily use the method of gradient descent to find the minimum which works great if the likelihood landscape is simple with a single minimum, but can fail in landscapes with multiple local extrema, depending on initial guesses of function parameters since the estimator might get stuck in a local minimum.
An alternative minimizer is the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). Evolution strategies (ESs) are continuous stochastic optimization algorithms searching for the minimum of a real valued function $f : \mathbb{R}^n \to \mathbb{R}$[2]. These strategies update (or evolve) their parameters with each iteration which allows for the ability to update their step size and search area to adapt to the function landscape.

Review

The algorithm of CMA-ES

The core idea behind CMA-ES starts with a parent distribution from which we sample λ points $\{x_1, ..., x_\lambda\}$ in the domain of the fitness function f. The parent distribution $\sigma\mathcal{N}(\vec{m}, C)$ is a multivariate normal distribution which is defined by its mean $\vec{m}$, its covariance matrix C and a step size (distribution scale) σ. We evaluate the fitness function only at these sampled points $\{f(x_1), ..., f(x_\lambda)\}$ and sort the sample points according to lowest fitness i.e. $\{x_1, ..., x_\lambda\} \xrightarrow{\text{sort by lowest fitness}} \{x_{1:\lambda}, ..., x_{\lambda:\lambda}\}$, where $x_{1:\lambda}$ denotes the sample value with the best (i.e. lowest) fitness. From the sorted sample we then select the μ best points (where $\mu \leq \lambda$) and then update the parent distribution and the parameters of the algorithm solely based on the μ best sorted values and repeat the process. The specific updates are given below [3]:

1. **Move the mean:** We move the mean (the placement) of the sample distribution towards the points with lowest fitness in $\{x_{1:\lambda}, ..., x_{\lambda:\lambda}\}$ using a weighted average where the best point is weighted greater than the next best which is greater than the third best and so on.

2. **Changing the shape:** The covariance matrix C defines the shape of the parent distribution $\sigma\mathcal{N}(\vec{m}, C)$. The best sample points alter the direction of the distribution towards the best fitness points, and also in the general direction of convergence. This is affected by the previous generations in a recursive manner with the oldest generations slowly losing their influence due to the learning rates c_μ and c_1.

3. **Control the step size:** If the previous generations all move towards a common general direction, we want to take larger steps (increase σ of the parent distribution) to optimize convergence. On the other hand, if the previous steps are canceling each other out, we might be oscillating around the minimum and thus we want to take smaller steps to increase precision.

This optimization continues until the algorithm exceeds the maximal number of generations or finds the minima either by seeing that the fitness doesn't change for multiple generations or the step size σ becomes so small that the algorithm deems the algorithm as having converged.

Application of the CMA-ES on the Six-hump camel function

To give a visual demonstration of the CMA-ES evolution, we have tested it on the Six-hump camel function:

$$f(x, y) = \left(4 - 2.1x^2 + \frac{x^4}{3}\right) x^2 + xy + (-4 + 4y^2)y^2 \tag{1}$$

We have used the 'cma-es' package for python[1], and see exactly what the algorithm explains. In Generation 1 of Figure 1 we have a very wide parent distribution where the red outlined points (μ best fitness points) are responsible for shifting the mean, update the covariance matrix, and reduce the step size as the algorithm closes in on one of the two global minima of the function. In Generation 12 we see that the parent distribution has converged to the global minimum.

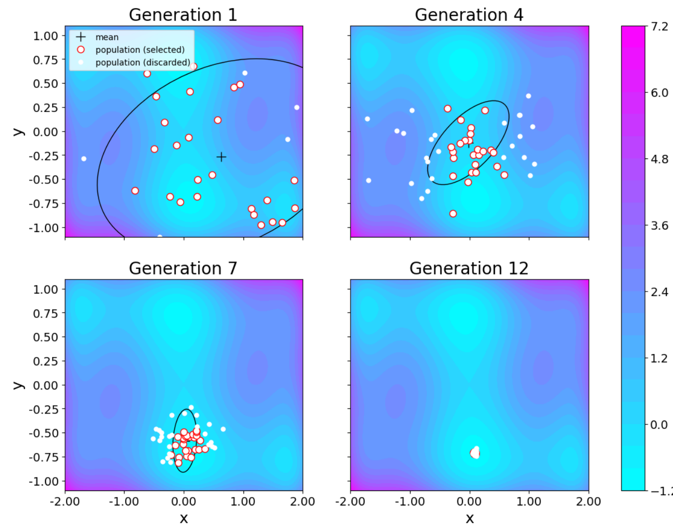

Figure 1: Numerical optimization of the Six-hump camel function using CMA-ES. The figure has 4 different generations showing the convergence to one of the global minima at (0.090, -0.713) [4]. Here the mean of the parent distribution is given by a black plus, the 50% confidence interval is given by the black ellipses and the points with red outline are the μ selected points with lowest fitness. Here the color bar also translates the function value to a color.

Besides the two global minima there are also two local minima in the outer corners of the domain. Even though CMA-ES contains many parameters that make it possible to find the global minima in rough function landscapes it unfortunately doesn't guarantee convergence since the points are sampled randomly and therefore can generate on top of a local minima. The risk of converging to a local minimum is greater when the sample size λ and initial step size σ are too small.

Conclusion

Although the algorithm is not always guaranteed to converge to the global extrema, it is still a very powerful tool for optimization. This is because it does not need a lot of information from the fitness-function (only a few direct function evaluations for each generation) and it is excellent at traversing bumpy landscapes. This comes in handy for MLE analysis, since the log-likelihood can be used as a fitness function. CMA-ES finds the minimum of $-\ln LLH(\theta)$ and thereby gives an estimate for the best fit parameters $\hat{\theta}$. CMA-ES also provides efficiency in larger dimensions, because it always samples λ points for each generation regardless of the dimensionality, in contrast to something like a raster scan which would go like $\sim N^{\text{Dimension}}$.

References

[1] cma-es. Python package, 2021. Version 0.9.2.

[2] Alexandre Chotard, Anne Auger, and Nikolaus Hansen. Cumulative step-size adaptation on linear functions, 2012.

[3] Nikolaus Hansen. The cma evolution strategy: A tutorial, 2023.

[4] M. Molga and C. Smutnicki. Test functions for optimization needs. Online, 2005. Retrieved 4th March 2025.