CMA-ES: COVARIANCE MATRIX ADAPTATION EVOLUTION STRATEGY

Luca Manzoni

CMA-ES

- CMA-ES is a multivariate EDA using a parametric distribution
- To represent the joint distribution CMA-ES uses a multivariate Gaussian, represented as a mean vector $\vec{m}$ and a covariance matrix C
- CMA-ES samples from the distribution and then uses the samples to update $\vec{m}$ and C
- For a more in-depth explanation we suggest
  Nikolaus Hansen, "*The CMA Evolution Strategy: A Tutorial*", 2016
  [https://arxiv.org/abs/1604.00772]

$(\mu/\mu_W, \lambda)$ CMA-ES

- Main idea:
  - Generate λ individuals from the current distribution with mean vector $\vec{m}$ and covariance matrix C
  - Keep the μ fittest individuals (truncated selection, since it is an evolution strategy)
  - Use the μ selected individuals to update $\vec{m}$ and C

COVARIANCE MATRIX REPRESENTATION

- To allow fast updates of C, CMA-ES does not actually store the covariance matrix directly (but we will ignore this detail after this slide)
- We can use the **eigendecomposition** (or **spectral decomposition**) of C as $C = B\Lambda B^{-1}$ where
  - B has the eigenvectors of C as its rows
  - Λ is a diagonal matrix with the eigenvalues of C
- Actually, since C is a symmetric, positive definite real matrix, we can write C as $BDDB^T$ where $DD = \Lambda$ and $B^T = B^{-1}$

SAMPLING INDIVIDUALS

- Individuals in CMA-ES are sampled from a multivariate Gaussian distribution with mean vector $\overrightarrow{m}$ and covariance matrix C…

- …and then are scaled my a mutation factor σ
  (in CMA-ES σ is called the **step size**)

→ NEED TO BE TOONED

- Higher values of σ means that the new individuals are more spread out than the actual distribution

REPRESENTATION OF INDIVIDUALS

- The same i^{th} individual can be represented in multiple (equivalent) ways:

- As a vector $\vec{y}^{(i)}$ sampled from a the distribution $N(\vec{0}, C)$, i.e., the distribution with covariance matrix C centred in the origin

- As a vector $\vec{x}^{(i)}$ as the vector $\vec{y}^{(i)}$ scaled by σ and shifted by the mean vector $\vec{m}$. That is, $\vec{x}^{(i)} = \vec{m} + \sigma\vec{y}^{(i)}$.
  This is the actual point for which the fitness is computed

CMA-ES: SIMPLE UPDATE

- A simple way to update $\vec{m}$ and C is to use the μ fittest individuals to recompute them:
  - $\vec{m} \leftarrow \frac{1}{\mu} \sum_{i=1}^{\mu} \vec{x}^{(i)}$
  - $C \leftarrow \frac{1}{\mu - 1} \sum_{i=1}^{\mu} \left(\vec{x}^{(i)} - \vec{m}\right) \left(\vec{x}^{(i)} - \vec{m}\right)^T$
- Instead of simply using the μ fittest individuals, we can add a weight to them in order to have the better ones with an higher influence on the update

CMA-ES: WEIGHTS

- Assume the individuals are ordered w.r.t. fitness with the best one being $\vec{x}^{(1)}$. Then $\overrightarrow{m}$ and C are updated as

- $\overrightarrow{m} \leftarrow \frac{1}{\mu} \sum_{i=1}^{\mu} w_i \vec{x}^{(i)}$

- $C \leftarrow \sum_{i=1}^{\mu} w_i \left(\vec{x}^{(i)} - \overrightarrow{m}\right) \left(\vec{x}^{(i)} - \overrightarrow{m}\right)^T$

- In CMA-ES the weight w_i is defined as $w_i = \ln\left(\frac{\lambda+1}{2i}\right) \Big/ \sum_{j=1}^{\mu} \ln\left(\frac{\lambda+1}{2j}\right)$

A DIFFERENT COVARIANCE MATRIX

- CMA-ES does not actually use the **updated** mean vector $\vec{m}$ to recompute C, but the old mean vector $\vec{m}_{\text{old}}$

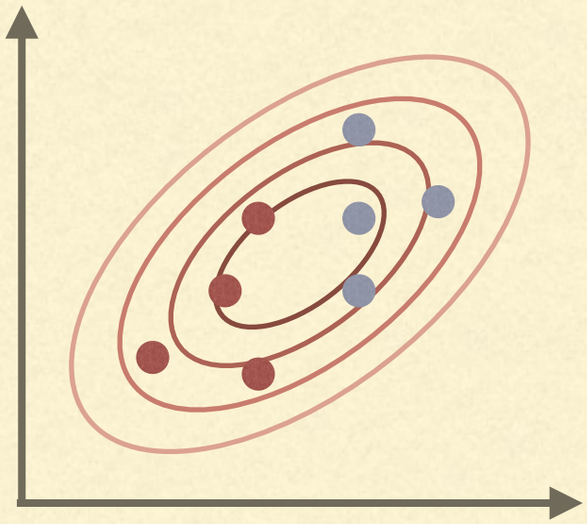
THE MAIN REASON IS TO AVOID A DISTRIBUTION TO CENTER TOO QUICKLY IN SOME SPACE

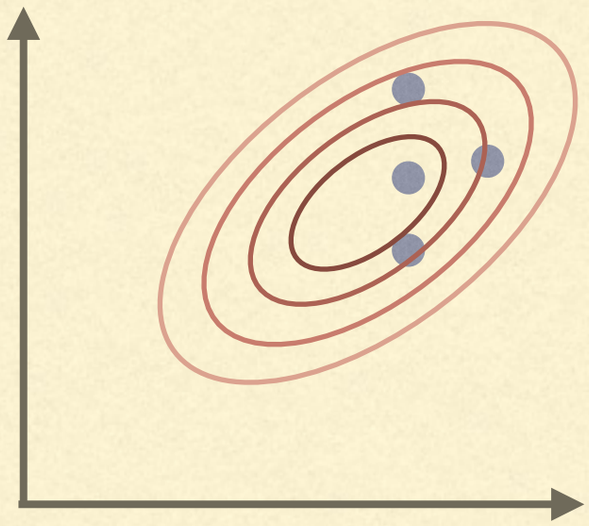
- $C \leftarrow \sum_{i=1}^{\mu} w_i \left(\vec{x}^{(i)} - \vec{m}_{\text{old}}\right)\left(\vec{x}^{(i)} - \vec{m}_{\text{old}}\right)^T$

- Recall that $\vec{x}^{(i)} = \vec{m}_{\text{old}} + \sigma\vec{y}^{(i)}$. We can simplify the update as

- $C \leftarrow \sigma^2 \sum_{i=1}^{\mu} w_i \vec{y}^{(i)}\vec{y}^{(i)T}$ and we can actually ignore the scaling factor σ^2

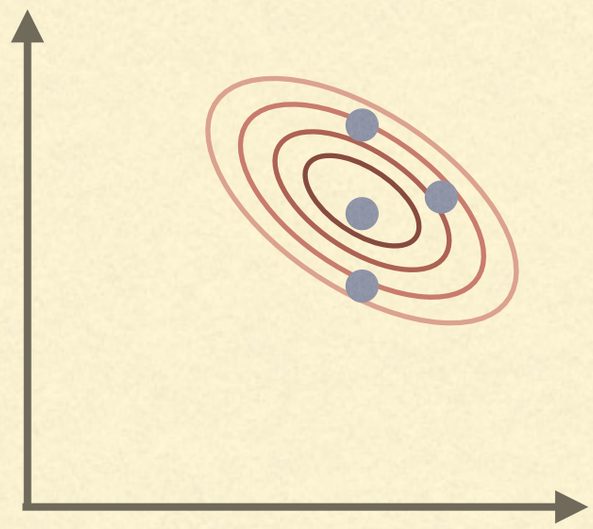
WHY TO USE THE OLD MEAN





Update with the old mean

We limit the risk of premature convergence



Update with the new mean

We risk to make the Gaussian too narrow and have premature convergence

GRADUAL UPDATES

- The mean vector $\overrightarrow{m}$ is re-computed at each time steps…
- ..but the covariance matrix C is only updated gradually:
  - $C \leftarrow (1 - c_\mu)C + c_\mu \sum_{i=1}^{\mu} \vec{y}^{(i)} \vec{y}^{(i)T}$
- This is called a **rank μ update**, since $\sum_{i=1}^{\mu} \vec{y}^{(i)} \vec{y}^{(i)T}$ is a rank μ matrix

AT MOST

THE EVOLUTION PATH

- CMA-ES also keep track of how the mean vector $\vec{m}$ changes with time

- In fact, CMA-ES keeps track of where $\vec{m}$ has been *historically heading* in the **evolution path** vector $\vec{p}$ which is updated at each time step as

  - $\vec{p} \leftarrow (1 - c_c)\vec{p} + c_c \frac{\vec{m} - \vec{m}_{\text{old}}}{\sigma}$ for a learning rate c_c

  - Actually, the updating rule for $\vec{p}$ is a little bit more involved, resulting in the actual evolution path $\vec{p}_c$

THE EVOLUTION PATH

- The evolution path $\vec{p}_c$ is used in the update of the covariance matrix as:

- $C \leftarrow (1 - c_1 - c_\mu)C + c_1(\vec{p}_c\vec{p}_c^T) + c_\mu \sum_{i=1}^{\mu} \vec{y}^{(i)}\vec{y}^{(i)^T}$

- Here $\vec{p}_c\vec{p}_c^T$ is a rank 1 matrix (hence the name c_1 for its coefficient) indicating the average direction in which the distribution has moved in the past

ADAPTIVE MUTATION RATE

- Another trick to make CMA-ES work is not to use a fixed mutation rate σ

- We can use the evolution path as a way to estimate how "fast" we are moving:
  - If the path is small the mutation rate will get smaller
  - If the path is large the mutation rate will get larger