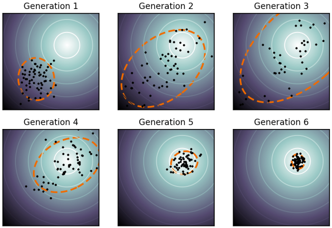
Optimization in Machine Learning

CMA-ES Algorithm



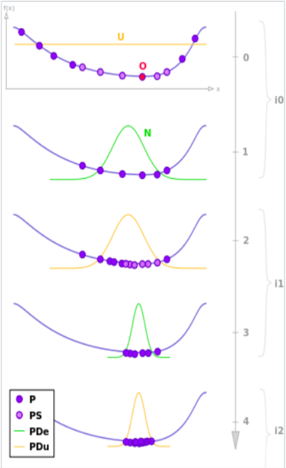
Learning goals

- CMA-ES strategy
- Estimation of distribution
- Step size control

CMA-ES

General algorithmic template with initial parameter setting $\boldsymbol{\theta}^{[0]}$ of parameterized density $p(\mathbf{x}|\boldsymbol{\theta})$, such that each iteration $t \in \{0, 1, \ldots, T\}$ consist of:

1. Draw λ samples, $\mathbf{x}^{(k)}$ from $p(\boldsymbol{x}|\boldsymbol{\theta}^{[t]})$
2. Evaluate $W(f(\mathbf{x}^{(k)}))$, where $W(\cdot)$ gives weights for each $\mathbf{x}^{(k)}$, typically 0 or 1 (order-preserving fitness transformation)
3. Find a $\boldsymbol{\theta}^{[t+1]}$ that uses weighted samples and corresponding function evaluations to move $p(\mathbf{x}|\boldsymbol{\theta})$ towards regions $\mathcal{S}$ that have large function values.



Estimation of distribution algorithm. For each iteration *i*, a random draw is performed for a population *P* in a distribution *PDu*. The distribution parameters *PDe* are then estimated using the selected points *PS*. The illustrated example optimizes a continuous objective function *f(X)* with a unique optimum *O*. The sampling (following a normal distribution *N*) concentrates around the optimum as one goes along unwinding algorithm.

CMA-ES

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is

- A state-of-the-art tool in evolutionary computation
- A stochastic/randomized method
- For usage in continuous domain
- For non-linear, non-convex optimization problems
- Useful in case “classical” search methods like quasi-Newton methods (BFGS) or conjugate gradient methods fail due to a non-convex or rugged search landscape (e.g. outliers, noise, local optima, sharp bends).

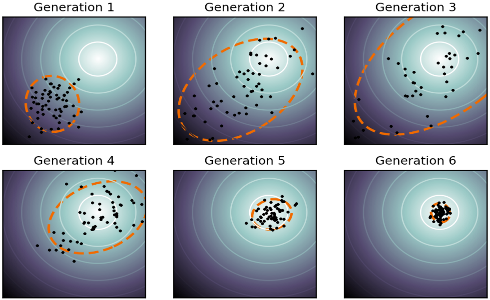
Detailed information on CMA-ES can be found in

1. Nikolaus Hansen. The CMA Evolution Strategy. 2005
2. A. Auger, N. Hansen: Tutorial CMA-ES: Evolution Strategies and Covariance Matrix Adaptation. 2012.

CMA-ES: STRATEGY

A population of new search points (individuals, offspring) is generated by sampling a multivariate normal distribution. A repeated update of the mean vector and covariance matrix with the respectively best ranked individuals moves the distributions towards the optimum.
Search points for generation number $t = 0, 1, \ldots, T$:

$$\mathbf{x}^{[t+1](k)} \sim \boldsymbol{m}^{[t]} + \sigma^{[t]}\mathcal{N}(\mathbf{0}, \boldsymbol{C}^{[t]}) \quad \text{for } k = 1, \ldots, \lambda.$$



CMA-ES: STRATEGY

$$\mathbf{x}^{[t+1](k)} \sim \boldsymbol{m}^{[t]} + \sigma^{[t]}\mathcal{N}(\mathbf{0}, \boldsymbol{C}^{[t]}) \quad \text{for } k = 1, \ldots, \lambda$$

- $\mathbf{x}^{[t+1](k)} \in \mathbb{R}^d$ is the k-th offspring (search point, individual) from generation $t+1$.
- $\lambda \geq 2$ is the population/sample size, number of offsprings.

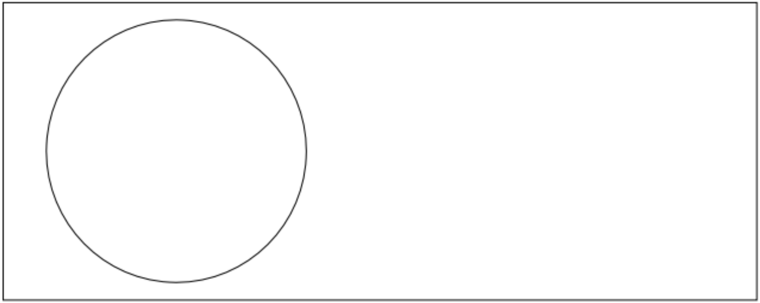
At generation t:

- $\mathcal{N}(\mathbf{0}, \boldsymbol{C}^{[t]})$ is multivariate normal distribution with zero mean, covariance matrix $\boldsymbol{C}^{[t]}$. *Note*: $\boldsymbol{m}^{[t]} + \sigma^{[t]}\mathcal{N}(\mathbf{0}, \boldsymbol{C}^{[t]}) \sim \mathcal{N}(\boldsymbol{m}^{[t]}, (\sigma^{[t]})^2\boldsymbol{C}^{[t]})$.
- $\boldsymbol{m}^{[t]} \in \mathbb{R}^d$ is the mean value of the search distribution.
- $\sigma^{[t]} \in \mathbb{R}_+$ is the "overall" standard deviation/step size.
- $\boldsymbol{C}^{[t]} \in \mathbb{R}^{d \times d}$ is the covariance matrix.

$\rightarrow$ *How to calculate* $\boldsymbol{m}^{[t+1]}$, $\boldsymbol{C}^{[t+1]}$, $\sigma^{[t+1]}$ *for next generation* $t+1$*?*

CMA-ES: BASIC METHOD - ITERATION 1

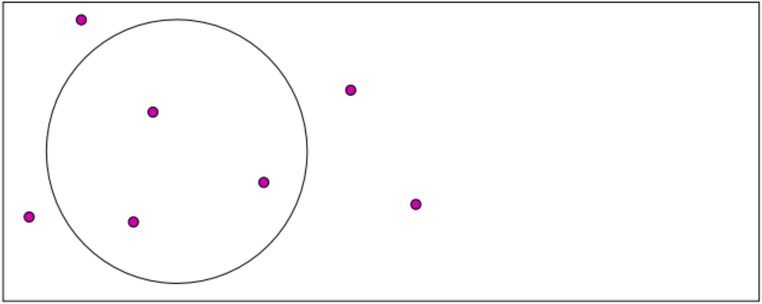
1. **Sample** from distribution
   $\mathbf{x}^{[1](k)} = \boldsymbol{m}^{[0]} + \sigma^{[0]}\mathcal{N}(\mathbf{0}, \boldsymbol{C}^{[0]})$ multivariate normal distribution.



Initial distribution $\mathcal{N}^{[0]} \sim (\mathbf{0}, \mathbb{I}_2)$ of generation $t = 0$.

CMA-ES: BASIC METHOD - ITERATION 1

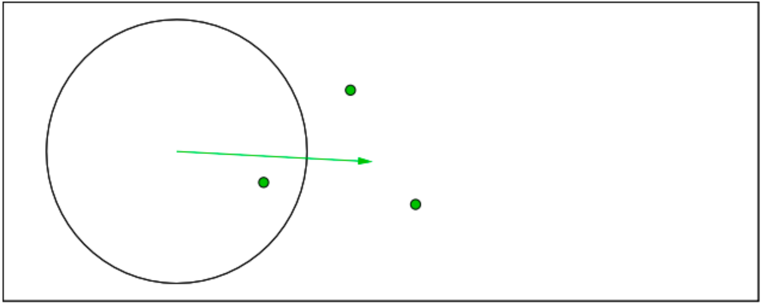
1. **Sample** from distribution
   $\mathbf{x}^{[1](k)} = \boldsymbol{m}^{[0]} + \sigma^{[0]}\mathcal{N}(\mathbf{0}, \boldsymbol{C}^{[0]})$ multivariate normal distribution.



Initial distribution $\mathcal{N}^{[0]} \sim (\mathbf{0}, \mathbb{I}_2)$ of generation $t = 0$, $\lambda = 6$.

CMA-ES: BASIC METHOD - ITERATION 1

2. **Ranking** solutions according to their fitness (*Selection* of μ best)
   $\mathbf{x}_{i:\lambda}$ as i-th ranked solution point, such that $f(\mathbf{x}_{1:\lambda}) \leq \cdots \leq f(\mathbf{x}_{\lambda:\lambda})$.

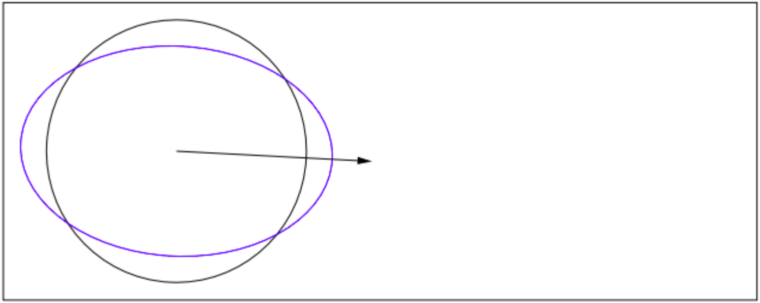


Calculation of auxiliary variable $\boldsymbol{y}_w := \sum_{i=1}^{\mu} w_i \mathbf{x}_{i:\lambda}$, using $\mu = 3$ selected points (high fitness $\rightarrow$ high weights)

CMA-ES: BASIC METHOD - ITERATION 1

3. **Update covariance matrix** (*Recombination*),

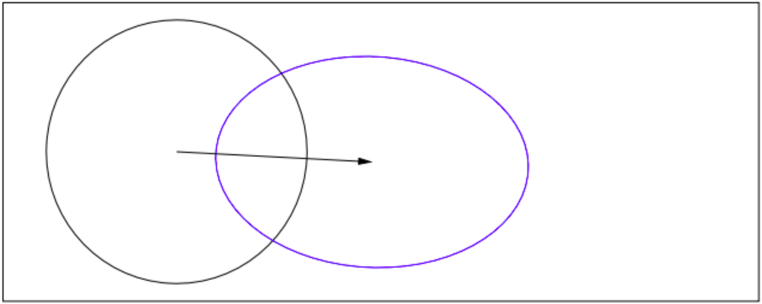
   improving "expected fitness" and likelihood for good steps.



Blue circle as a mixture of $\boldsymbol{C}$ and step $\boldsymbol{y}_w$ (simplified):
$\boldsymbol{C}^{[1]} = 0.8\boldsymbol{C}^{[0]} + 0.2\boldsymbol{y}_w^{[0]}(\boldsymbol{y}_w^{[0]})^\top$ (Rank 1 update).

CMA-ES: BASIC METHOD - ITERATION 1

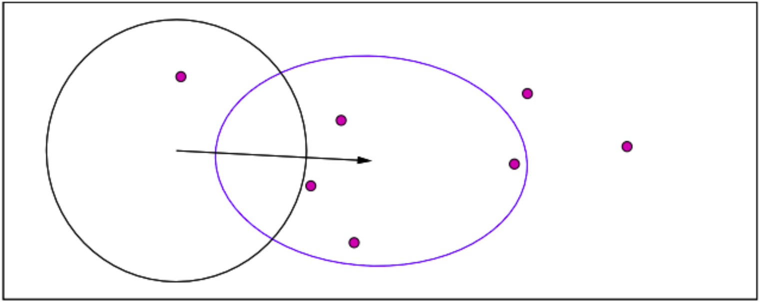
4. **Update mean**



Movement towards the new distribution with mean
$\boldsymbol{m}^{[1]} = \boldsymbol{m}^{[0]} + \sigma^{[0]} \boldsymbol{y}_w^{[0]}$.

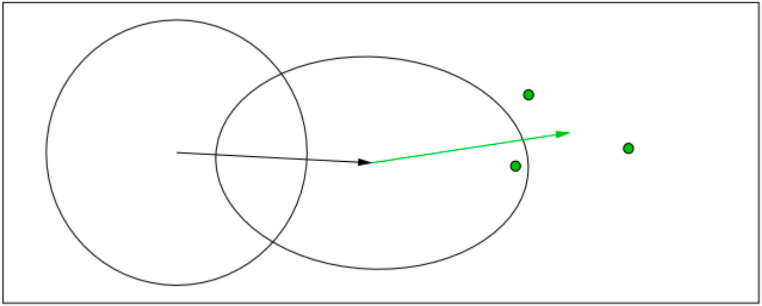
CMA-ES: BASIC METHOD - ITERATION 2

1. **Sample** from distribution for new generation



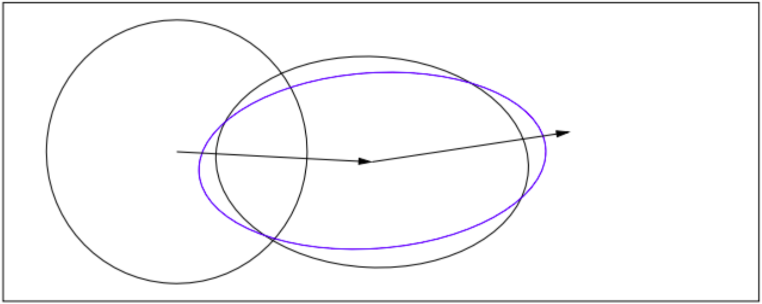
CMA-ES: BASIC METHOD - ITERATION 2

2. **Ranking** solutions according to their fitness (*Selection* of μ best)



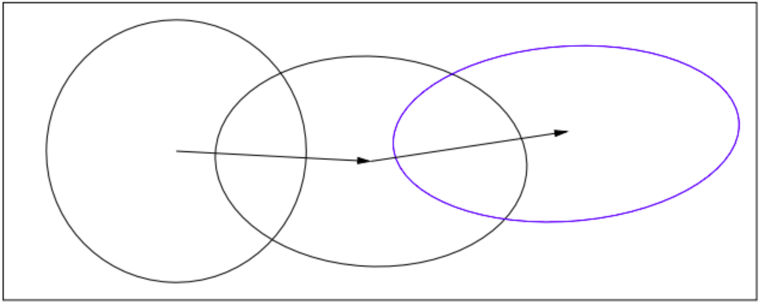
CMA-ES: BASIC METHOD - ITERATION 2

3. **Update mean and covariance matrix** (*Recombination*)



CMA-ES: BASIC METHOD - ITERATION 2

4. **Update step-size** based on non-local information, exploit correlations in the history of steps.



UPDATING C: CMA - RANK-ONE UPDATE

Initialize $\boldsymbol{m} \in \mathbb{R}^d$ and $\boldsymbol{C} = \mathbb{I}$, set $\sigma = 1$, learning rate $c_{cov} \approx 2/d^2$.
While not terminate

$$\mathbf{x}^{(k)} = \boldsymbol{m} + \sigma \mathcal{N}_i(\mathbf{0}, \boldsymbol{C})$$

$$\boldsymbol{m} \leftarrow \boldsymbol{m} + \sigma \boldsymbol{y}_w, \quad \text{where } \boldsymbol{y}_w = \sum_{i=1}^{\mu} \boldsymbol{w}_i \mathbf{x}_{i:\lambda}$$

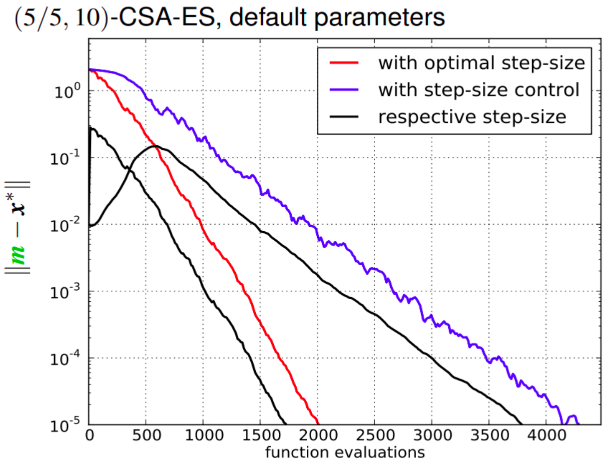
$$\boldsymbol{C} \leftarrow (1 - c_{cov})\boldsymbol{C} + c_{cov}\mu_w \underbrace{\boldsymbol{y}_w \boldsymbol{y}_w^\top}_{\text{rank-one}}, \quad \text{where } \mu_w = \frac{1}{\sum_{i=1}^{\mu} w_i^2}$$

The rank-one update was developed in several domains independently, conducting a **principle component analysis** (PCA) of steps $\boldsymbol{y}_w$ sequentially in time and space.

UPDATING σ: METHODS STEP-SIZE CONTROL

- $1/5$**-th success rule**: increases the step-size if more than 20 % of the new solutions are successful, decrease otherwise
- σ**-self-adaptation**: mutation is applied to the step-size and the better - according to the objective function value - is selected
- **Path length control via cumulative step-size adaptation (CSA)**: self-adaptation derandomized and non-localized
- Alternative step-size adaptation mechanism: two-point step-size adaptation, median success rule, population success rule.

UPDATING σ: PATH LENGTH CONTROL (CSA)



$$f(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2$$

in $[-0.2, 0.8]^n$
for $n = 30$

CSA effective and robust for $\lambda \leq N$.