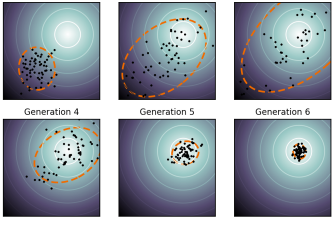
Optimization in Machine Learning

Evolutionary Algorithms
CMA-ES Algorithm



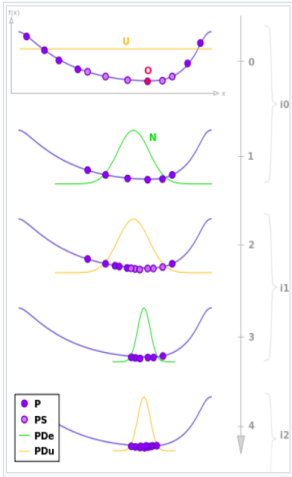
Learning goals

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

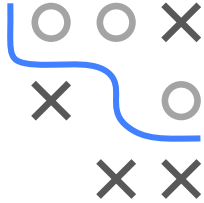
ESTIMATION OF DISTRIBUTION ALGORITHM

- Instead of population, maintain distribution to sample offspring from

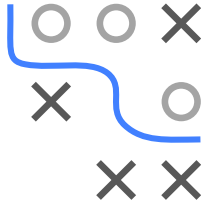
1. Draw λ offsprings $\mathbf{x}^{(i)}$ from $p(\cdot|\boldsymbol{\theta}^{[t]})$
2. Evaluate fitness $f(\mathbf{x}^{(i)})$
3. Update $\boldsymbol{\theta}^{[t+1]}$ with μ best offsprings



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.



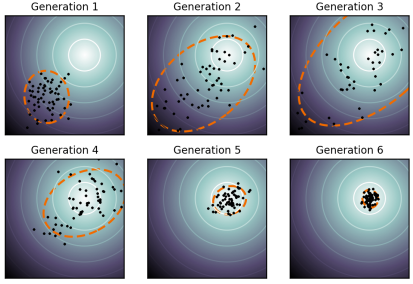
COVARIANCE MATRIX ADAPTATION



Sample distribution is multivariate Gaussian

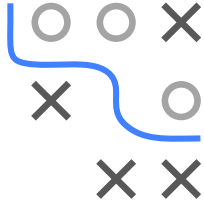
$$\mathbf{x}^{[t+1](i)} \sim \mathbf{m}^{[t]} + \sigma^{[t]}\mathcal{N}(\mathbf{0}, \mathbf{C}^{[t]}) \quad \text{for } i = 1, \ldots, \lambda$$

- $\mathbf{x}^{[t+1](i)} \in \mathbb{R}^d$ i-th offspring; $\lambda \geq 2$ number of offspring
- $\mathbf{m}^{[t]} \in \mathbb{R}^d$ mean value and $\mathbf{C}^{[t]} \in \mathbb{R}^{d \times d}$ covariance matrix
- $\sigma^{[t]} \in \mathbb{R}_+$ "overall" standard deviation/step size



Question: How to adapt $\mathbf{m}^{[t+1]}$, $\mathbf{C}^{[t+1]}$, $\sigma^{[t+1]}$ for next generation $t + 1$?

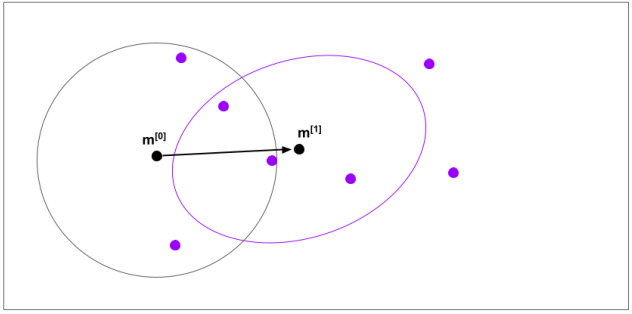
CMA-ES: BASIC METHOD - ITERATION 1

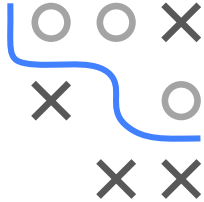


0. Initialize $\mathbf{m}^{[0]}, \sigma^{[0]}$ problem-dependent and $\mathbf{C}^{[0]} = \mathbf{I}_d$

CMA-ES: BASIC METHOD - ITERATION 2

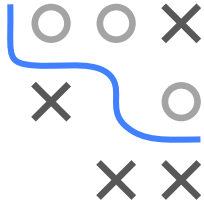
1. **Sample** from distribution for new generation





UPDATING C: FULL UPDATE

Full CMA update of $\mathbf{C}$ combines rank-μ update with a rank-1 update using exponentially smoothed evolution path $\mathbf{p}_c \in \mathbb{R}^d$ of successive steps and learning rate c_1:



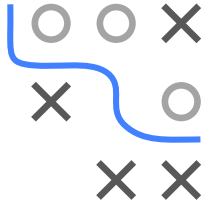
$$\mathbf{p}_c^{[0]} = \mathbf{0}, \quad \mathbf{p}_c^{[t+1]} = (1 - c_1)\mathbf{p}_c^{[t]} + \sqrt{\frac{c_1(2 - c_1)}{\sum_{i=1}^{\mu} w_i^2}}\mathbf{y}_w$$

Final update of $\mathbf{C}$ is

$$\mathbf{C}^{[t+1]} = (1 - c_1 - c_\mu \textstyle\sum_j w_j)\mathbf{C}^{[t]} + c_1 \underbrace{\mathbf{p}_c^{[t+1]}(\mathbf{p}_c^{[t+1]})^\top}_{\text{rank-1}} + c_\mu \underbrace{\sum_{i=1}^{\mu} w_i \mathbf{y}_{i:\lambda}^{[t+1]}(\mathbf{y}_{i:\lambda}^{[t+1]})^\top}_{\text{rank-}\mu}$$

- Correlation between generations used in rank-1 update
- Information from entire population is used in rank-μ update

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)**
  Intuition:
  - Short cumulative step-size $\triangleq$ steps cancel $\rightarrow$ decrease $\sigma^{[t+1]}$
  - Long cumulative step-size $\triangleq$ corr. steps $\rightarrow$ increase $\sigma^{[t+1]}$