

CMAES

Covariance Matrix Adaptation Evolution Strategy

Opracowanie: Lidia Wojciechowska

W algorytmie CMAES, podobnie jak w algorytmie EDA, adaptowany jest rozkład prawdopodobieństwa generacji punktów, opisany przez wartość oczekiwaną oraz macierz kowariancji, a także, dodatkowo, przez parametr długości kroku. Dwa najważniejsze mechanizmy wykorzystywane w tym algorytmie to:

- adaptacja macierzy kowariancji;
- adaptacja długości kroku.

Zapis algorytmu

$$C(1) = I, p_c(1) = 0, p_\sigma(1) = 0$$

while ! stop

generuj $d_i(t) \sim N(0, C(t)), i = 1 \dots \lambda$

oblicz $q_i(t) = q(m(t) + \sigma(t) \cdot d_i(t))$

sortuj według $q_i(t)$

$$\Delta(t) = \frac{1}{\mu} \sum_{i=1}^{\mu} d_i(t)$$

$$m(t+1) = m(t) + \sigma(t) \cdot \Delta(t)$$

$$p_\sigma(t+1) = (1 - c_\sigma)p_\sigma(t) + C^{-1/2} \sqrt{1 - (1 - c_\sigma)^2} \sqrt{\mu} \Delta(t)$$

$$p_c(t+1) = (1 - c_c)p_c(t) + \sqrt{1 - (1 - c_c)^2} \sqrt{\mu} \Delta(t)$$

$$\sigma(t+1) = \sigma(t) \cdot \exp\left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|p_\sigma\|}{E\|N(0, I)\|} - 1\right)\right)$$

$$C(t+1) = (1 - c_1 - c_\mu)C(t) + c_1 p_c(t+1)p_c(t+1)^T + c_\mu \sum_{i=1}^{\mu} d_i(t)d_i(t)^T$$

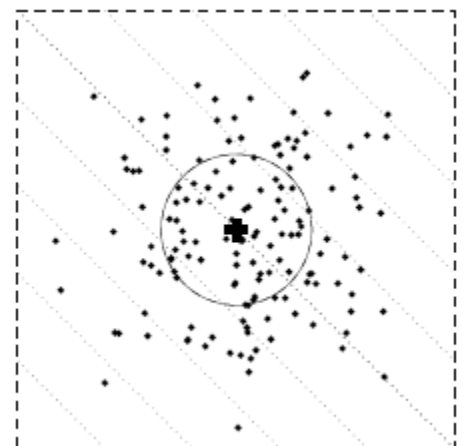
$t \leftarrow t + 1$

Generacja populacji oraz selekcja

Populacja $q_i(t)$ w kroku t generowana jest w następujący sposób:

1. Generowane jest λ punktów $d_i(t)$ z rozkładu normalnego o zerowej wartości oczekiwanej i zadanej macierzy kowariancji $C(t)$.
2. Faktyczna populacja $q_i(t)$ tworzona jest na podstawie punktów $d_i(t)$ poprzez pomnożenie ich przez współczynnik długości kroku $\sigma(t)$ i przesunięcie o wartość oczekiwaną osobnika populacji $m(t)$. Wyjaśnienie, czym dokładnie są te wielkości, podane jest w dalszej części niniejszego rozdziału.

Z populacji wybierane jest μ najlepszych punktów – w praktyce może to być np. połowa wygenerowanych wcześniej punktów.



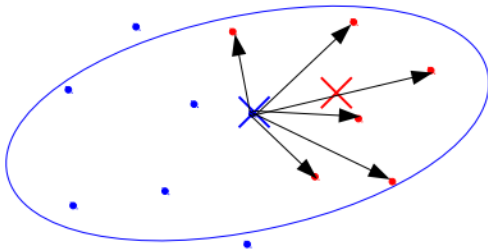
Adaptacja macierzy kowariancji (CMA)

Dla wygody przyjmijmy, że długość kroku, czyli $\sigma = 1$. Przypomnijmy, że macierz kowariancji jest uogólnieniem wariancji na przypadek wielowymiarowy, czyli w praktyce charakteryzuje rozproszenie punktów populacji.

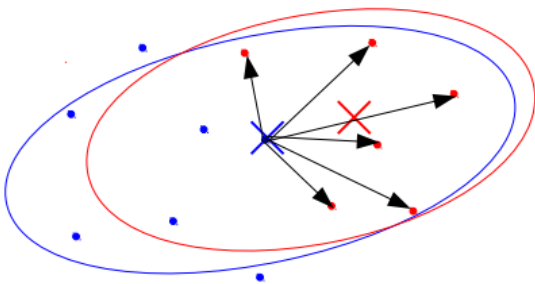
Adaptacja macierzy kowariancji w algorytmie CMAES różni się od tejże adaptacji w algorytmie EDA tym, że od punktów nie odejmujemy nowej wartości średniej a starą wartość oczekiwaną: $m(t)$. Można to zapisać w następujący sposób:

$$C(t+1) = \frac{1}{\mu} \sum_i (P_i(t) - m(t))(P_i(t) - m(t))^T$$

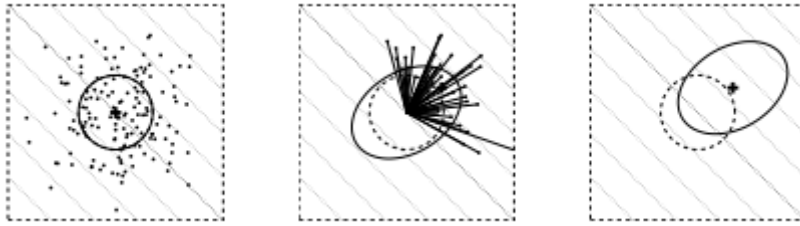
Oznacza to, że podczas tworzenia nowej macierzy kowariancji od wyselekcjonowanych μ punktów populacji odejmujemy wartość oczekiwaną populacji przed selekcją. Zilustrowane jest to poniższym rysunkiem:



Punkty niebieskie i czerwone to populacja bazowa, której rozkład przybliżony jest niebieską elipsą, i którego wartość oczekiwana znajduje się w punkcie oznaczonym niebieskim krzyżykiem. Punkty czerwone to wyselekcjonowana część populacji – ich wartość oczekiwana to czerwony krzyżyk. Jako środek populacji, wykorzystany do stworzenia nowej macierzy kowariancji, wykorzystany jest punkt oznaczony krzyżykiem niebieskim. Powstały dzięki temu nowy rozkład przybliżony jest na poniższym rysunku czerwoną elipsą.



Dzięki tej metodzie adaptacji macierzy kowariancji kształt elipsy opisującej rozkład będzie „wyciągał się” w stronę zwiększającej się wartości funkcji celu, układając się prostopadłe do poziomicy tej funkcji, zamiast, jak w przypadku algorytmu EDA, rozszerzać się równoległe do nich. Oznacza to, że na zboczach funkcji celu algorytm CMAES będzie zachowywał się bardzo dobrze, dążąc szybko w stronę optimum. Poniższy rysunek pokazuje, jak rozkład populacji adaptuje się w przypadku funkcji celu o wartościach rosnących w stronę prawego górnego rogu:



Kiedy populacja znajdzie się „na szczycie”, czyli w optimum funkcji celu, wartość oczekiwana wybranych punktów i wartość oczekiwana populacji bazowej (czerwony i niebieski krzyżyk) będą się w przybliżeniu pokrywały - rozkład będzie się więc zachowywał jak w przypadku algorytmu EDA, czyli zacznie się skracać i zmniejszać, pokrywając wciąż optimum.

Wzór opisujący adaptację macierzy kowariancji na początku niniejszego rozdziału jest jednak nieco bardziej skomplikowany. Dlaczego?

Po pierwsze, algorytm wykorzystuje do adaptacji macierzy kowariancji informacje o korelacji między kolejnymi krokami. Parametr p_c to tzw. ścieżka ewolucji – akumulacja kolejnych kroków wykonanych przez algorytm. Pierwszy składnik poniższego równania określa, z jaką wagą wykorzystamy dotychczasową ścieżkę ewolucyjną, a drugi to znormalizowany ostatni krok.

$$p_c(t + 1) = (1 - c_c)p_c(t) + \sqrt{1 - (1 - c_c)^2} \sqrt{\mu} \Delta(t)$$

Ścieżka ta, czyli informacja o korelacji między kolejnymi generacjami populacji, wykorzystana jest (z wagą c_1) w drugim składniku równania opisującego aktualizację macierzy kowariancji. Istotne jest to szczególnie w przypadku małych populacji.

Ponadto algorytm skutecznie wykorzystuje informacje o wszystkich punktach z populacji – jest to szczególnie ważne w przypadku dużych populacji. Odpowiedzialny za to jest (z wagą c_μ) trzeci składnik poniższego równania. Pierwszy składnik zaś jest uwzględnieniem z pewną wagą poprzedniej macierzy kowariancji.

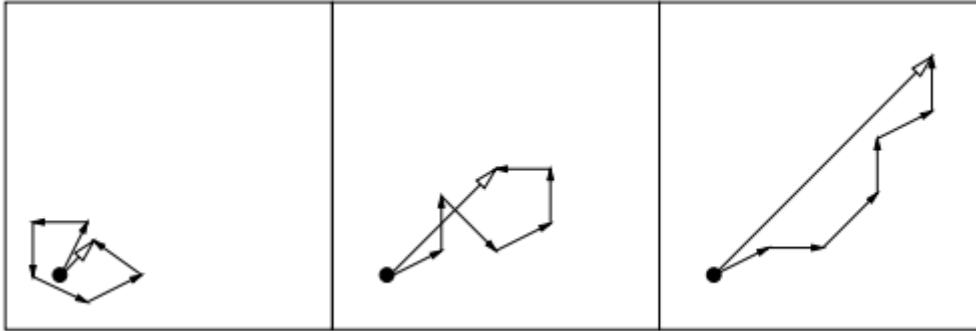
$$C(t + 1) = (1 - c_1 - c_\mu)C(t) + c_1 p_c(t + 1) p_c(t + 1)^T + c_\mu \sum_{i=1}^{\mu} d_i(t) d_i(t)^T$$

Adaptacja długości kroku (CSA)

Krokiem będziemy nazywać różnicę między wartością oczekiwaną rozkładów dwóch kolejnych populacji (czyli w odniesieniu do powyższych rysunków - przesunięcie kolejnych elips). Innymi słowy wartość współczynnika długości kroku odpowiada za to, jak daleko punkt wygenerowany może odsunąć się od punktu środkowego.

Przyjmijmy dla wygody, że macierz kowariancji jest jednostkowa.

Kroki oznaczone są na poniższym rysunku strzałkami.



Pierwszy przypadek od lewej to sytuacja, w której długość kroku jest za duża – strzałki o zamalowanych grotach krążą wokół pewnego punktu, ich kierunki znoszą się wzajemnie. Taka sytuacja może wystąpić na szczycie funkcji celu – zbyt długi krok powoduje, że punkty trafiają za daleko od optimum, „przestrzeliwują się” na drugą stronę szczytu. Krok należy wtedy skrócić – skrócony krok to na rysunku strzałka o niezamalowanym grotcie.

Przypadek po prawej stronie to sytuacja, w której kroki są za krótkie. Strzałki o niezamalowanych grotach wskazują w przybliżeniu podobny kierunek – gdyby krok został wydłużony, ten sam rezultat można by osiągnąć wykonując mniejszą liczbę kroków - skróciłoby to czas działania algorytmu.

Sytuacja, do której chcemy dążyć, pokazana jest na środkowym obrazku. Kroki są w przybliżeniu prostopadłe do siebie, nieskorelowane ze sobą.

Adaptacja długości kroku to aktualizacja parametru σ . Aby zdecydować, czy krok ma być zmniejszony, zwiększony, czy pozostać niezmienny, porównujemy długość wektora p_σ z wartością oczekiwaną rozkładu normalnego standaryzowanego. Jeżeli wartości te będą równe, wtedy $\sigma(t + 1)$ obliczona z poniższego wzoru będzie równa $\sigma(t)$.

$$\sigma(t + 1) = \sigma(t) \cdot \exp\left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|p_\sigma\|}{E\|N(0, I)\|} - 1\right)\right)$$

Czym jest wektor p_σ ? Możemy nazwać go sprzężoną ścieżką ewolucji, tak jak w przypadku adaptacji macierzy kowariancji akumuluje on wektory przesunięcia punktu środkowego populacji. Jeżeli ruch będzie odbywał się wciąż w tę samą stronę, długość tego wektora będzie większa od $E\|N(0, I)\|$, a jeżeli będą następowały fluktuacje na wznęgu, będzie ona mniejsza. Dlatego też w pierwszym przypadku długość kroku zostanie zwiększona, a w drugim zmniejszona. Wektor p_σ aktualizowany jest w następujący sposób:

$$p_\sigma(t + 1) = (1 - c_\sigma)p_\sigma(t) + C^{-1/2}\sqrt{1 - (1 - c_\sigma)^2}\sqrt{\mu}\Delta(t)$$

c_σ to parametr określający, jaką część informacji o poprzednim wektorze $p_\sigma(t)$ chcemy zachować. Drugi składnik równania jest znormalizowanym wektorem ostatniego przesunięcia.

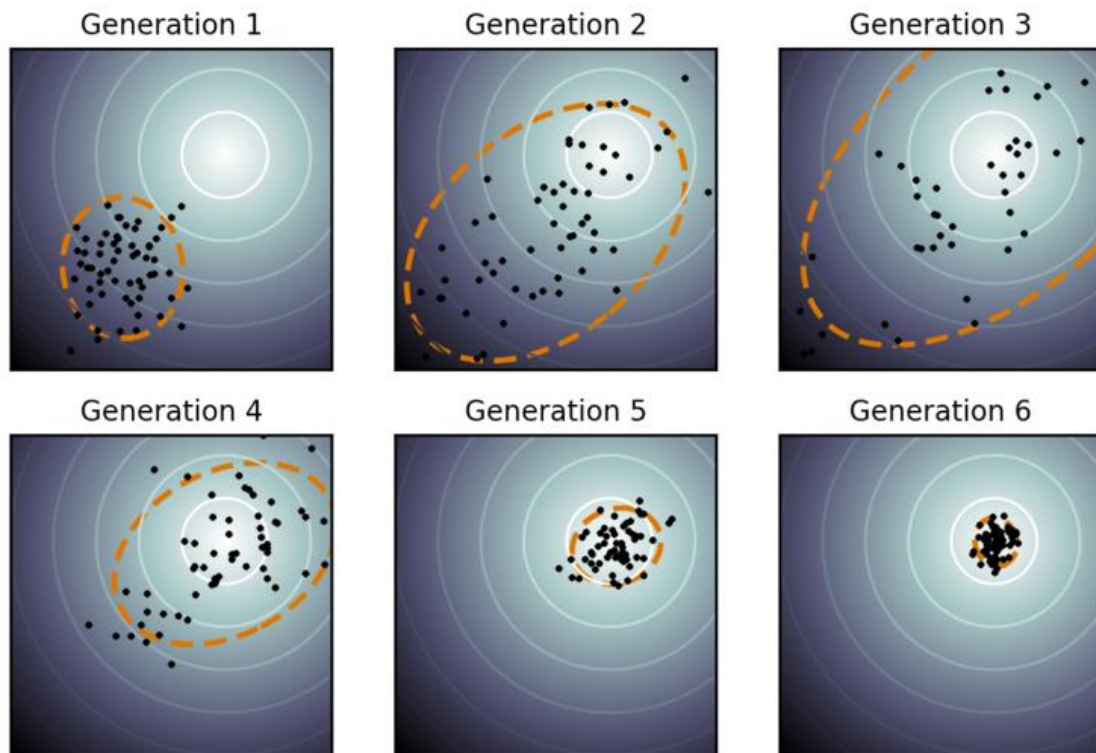
Całość algorytmu

1. Generacja i selekcja punktów – jak opisano powyżej.
2. Obliczana jest nowa wartość średnia wybranych μ punktów, czyli rodziców: $m(t + 1)$ – wartość ta jest „przesunięta” względem starej wartości średniej w stronę lepszych punktów wybranych z populacji.
3. Aktualizowane są parametry p_σ oraz p_c .
4. Następuje aktualizacja długości kroku σ , jak opisano powyżej.

5. Aktualizowana jest macierz kowariancji, jak opisano powyżej.

Warunek końca algorytmu zależy od specyfiki zadania.

Na poniższym rysunku przedstawiony jest przykładowy przebieg algorytmu CMAES. Kształt macierzy kowariancji adaptuje się na podstawie selekcji punktów z populacji, a skala tej macierzy zależy od ścieżki ewolucji.



Źródła

1. Prof. J. Arabas - Slajdy z wykładów ALHE.
2. Nikolaus Hansen - *The CMA Evolution Strategy: A Tutorial*
3. <https://en.wikipedia.org/wiki/CMA-ES>, dostęp: 22.01.2019.