1. The SPSA "process"

We analyze an elegant chess oriented version of the SPSA algorithm invented by Stockfish author Joona Kiiski [3].

Let $(\mathfrak{c}_i)_{i\geq 0} \in \mathbb{R}^n$ be a sequence of (column) vectors and let $(r_i)_{i\geq 0} \in \mathbb{R}^n_{>0}$. To go from a vector of engine parameters $\theta_i = (\theta_i^{(j)})_{j=1,\dots,n} \in \mathbb{R}^n$ to a new vector $\theta_{i+1} \in \mathbb{R}^n$ we play a match of $\theta_i + \mathfrak{c}_i$ against $\theta_i - \mathfrak{c}_i$ and put

$$\theta_{i+1} = \begin{cases} \theta_i + r_i\mathfrak{c}_i & \text{in case of a win} \\ \theta_i & \text{in case of a draw} \\ \theta_i - r_i\mathfrak{c}_i & \text{in case of a loss} \end{cases} \tag{1.1}$$

Since we have biased[1] the update of θ_i in the direction of the engine that won the game, the hope is that θ_i for $i \gg 0$ will yield a stronger engine than θ_0.

The next two examples illustrate standard choices for $\mathfrak{c}$.

Example 1.1. Let $c_0, \dots, c_{n-1} > 0$ and put $\mathfrak{c}_i = c_{\bar{\imath}}e_{\bar{\imath}}$ where $\bar{\imath} = i \bmod n$ and e_i is the i'th basis vector of $\mathbb{R}^n$. This is a variant of the FDSA algorithm. We have

$$E(\mathfrak{c}\mathfrak{c}^t) = \frac{1}{n}\operatorname{diag}(c_1^2, \dots, c_n^2)$$

Example 1.2. Let $c_0, \dots, c_{n-1} > 0$ and put $\mathfrak{c}_i = \sum_i c_i e_i \Delta_i$ where Δ_i is chosen uniform randomly in $\{\pm 1\}$. Then

$$E(\mathfrak{c}\mathfrak{c}^t) = \operatorname{diag}(c_1^2, \dots, c_n^2)$$

This is the SPSA algorithm.

Assumption 1.3. *We allow $(\mathfrak{c}_i)_i$ to be stochastic but throughout we assume that $(\mathfrak{c}_i\mathfrak{c}_i^t)_i$ stays close to its short term running average/expectation value (denoted by $E(\mathfrak{c}^t\mathfrak{c})$) and that the latter is an invertible $n \times n$ matrix.*

2. Main results

2.1. **Statements.** Let e be the function which maps $\theta \in \mathbb{R}^n$ to Elo. Assume that e is (approximately) quadratic and attains its maximum value at $\theta = \bar{\theta}$. We normalize e such that $e(\bar{\theta}) = 0$.

Apply the generalized SPSA algorithm with constant r and with $\mathfrak{c}$ satisfying Assumption 1.3 such that $E(\mathfrak{c}\mathfrak{c}^t)$ is constant. We will heuristically drive some reasonable approximations.

(1) Asymptotically the expectation value of $e(\theta_i)$ will satisfy[2]

$$\boxed{E(e(\theta_i)) = -\frac{Cn(1-d)}{8}r} \tag{2.1}$$

for

$$C = 800/\ln(10) = 347.43558552.$$

[1] For the purpose of exposition we assume that the games are played with an opening book that is perfectly balanced. In practice this will never be satisfied but one may simulate it by replaying every game with reversed colors. This will be assumed silently below.

[2] With an unbalanced book and replayed color reversed games the factor $1-d$ in the formula can be reduced to $1-d-4b$ where b is the RMS bias of the opening book (computed using $0, 1/2, 1$ scoring, see [6]). The same holds true for similar formulas below.

(2) Let $0 \leq \gamma \leq 1$ and define z by $\gamma = P(X \leq z)$ for $X \sim \chi^2_n$. E.g. for $\gamma = 0.95$ we have in low dimensions

dim	z
1	3.8414588206941236
2	5.9914645471079799
3	7.8147279032511765
4	9.4877290367811540

With probablity γ, $e(\theta_i)$ will asymptotically satisfy

$$\boxed{e(\theta_i) \geq -\frac{Cz(1-d)}{8}r} \tag{2.2}$$

with $d \in [0,1]$ being equal to the draw ratio

(3) The expectation value of θ_i will converge linearly towards $\bar{\theta}$ as

$$\sim e^{-i/\lambda}$$

with the time constant λ bounded by

$$\boxed{\lambda = \frac{C}{2r\mu}} \tag{2.3}$$

where μ is the smallest eigenvalue of [3] $-E(\mathfrak{c}^t\mathfrak{c})\operatorname{Hess}(e)$.

Remark 2.1. We see from (3) that the nicest possible case is when

$$-E(\mathfrak{c}^t\mathfrak{c})\operatorname{Hess}(e) = \mu I_{n\times n}.$$

It follows from the derivation below that in that case the update (1.1) is (on average) proportional to $r\mu \operatorname{Hess}(e)^{-1}(\operatorname{grad} e)(\theta_i)$, which is in particular the optimal direction for gradient descent.

2.2. **Derivations.** Below vectors are always column vectors. We will also switch to continuous time. Hence we write $\theta(t)$ (or simply θ) instead of θ_i, etc...

We can think of the dynamics of θ as a discrete version of the continuous process

$$d\theta = r\mathfrak{c}(\mu_{\theta,\mathfrak{c}}dt + \sigma_{\theta,\mathfrak{c}}dW_t)$$

where $(\mu_{\theta,\mathfrak{c}}, \sigma_{\theta,\mathfrak{c}})$ are the mean and standard deviation of a single game between parameters $\theta \pm \mathfrak{c}$ (scored as $-1, 0, 1$) and W_t is a Wiener process (Brownian motion). We simplify further and assume that $\sigma_{\theta,\mathfrak{c}}$ is independent of $\theta, \mathfrak{c}$ so that

$$\sigma_{\theta,\mathfrak{c}} := \sigma \cong 1 - d \tag{2.4}$$

where d is the draw ratio[4].

Let $f(\theta)$ be the function that maps θ to the expected score against a reference engine (scored as $-1, 0, 1$). Assuming linearity we get

$$\mu_{\theta,\mathfrak{c}} \cong \operatorname{score}(\theta + \mathfrak{c}, \theta - \mathfrak{c}) \cong f(\theta + \mathfrak{c}) - f(\theta - \mathfrak{c}) = 2\mathfrak{c}^t(\operatorname{grad} f)(\theta)$$

we find

$$d\theta = 2r\mathfrak{c}\mathfrak{c}^t(\operatorname{grad} f)(\theta)dt + r\sigma\mathfrak{c}dW_t$$

[3] In the formula we have used that e is quadratic so that $\operatorname{Hess}(e)(\theta)$ does not depend on θ.

[4] See Footnote 2.

We now replace $\mathfrak{c}\mathfrak{c}^t$ by its short term average/expectation value which we denote by $E(\mathfrak{c}\mathfrak{c}^t)$ and we put $\mathfrak{a} = rE(\mathfrak{c}\mathfrak{c}^t)$. Note that $\mathfrak{a}$ is symmetric. We obtain

$$d\theta = 2\mathfrak{a}(\operatorname{grad} f)(\theta)dt + r\sigma\mathfrak{c}dW_t$$

The conversion between f and e is approximately given by

$$f(\theta) = e(\theta)/C \tag{2.5}$$

So f is approximately quadratic as well. Hence (translating such that $\bar{\theta} = 0$)

$$f(\theta) = -\frac{1}{2}\theta^t \Xi \theta$$

with Ξ a positive definite symmetric $n \times n$-matrix. So

$$(\operatorname{grad} f)(\theta) = -\Xi\theta$$

from which we get

$$d\theta = -2\mathfrak{a}\Xi\theta dt + r\sigma\mathfrak{c}dW_t$$

The solution of the corresponding homogeneous equation

$$d\theta = -2\mathfrak{a}\Xi\theta dt$$

is given by

$$\theta = \exp(-2\mathfrak{a}\Xi t)D$$

where D is a column vector of constants. We now solve the inhomogeneous system using variation of constants (i.e. we make D depend on t). We get

$$\exp(-2\mathfrak{a}\Xi t)D'(t) = r\sigma\mathfrak{c}dW_t$$

$$D(t) = C + r\sigma\int_0^t \exp(2\mathfrak{a}\Xi s)\mathfrak{c}dW_s$$

So

$$\theta(t) = \exp(-2\mathfrak{a}\Xi t)\theta(0) + r\sigma\int_0^t \exp(2\mathfrak{a}\Xi(s-t))\mathfrak{c}dW_s$$

So $\theta(t)$ is multivariate normal with expectation value

$$E(\theta(t)) = \exp(-2\mathfrak{a}\Xi t)\theta(0) = \exp(2\mathfrak{a}\operatorname{Hess}(f)t)\theta(0)$$

from which we deduce (4,3) using the conversion (2.5) combined with (2.4).

We now calculate the covariance of $\theta(t)$.

$$\operatorname{Cov}(\theta(t)) = r^2\sigma^2\int_0^t \exp(2\mathfrak{a}\Xi(s-t))\mathfrak{c}\mathfrak{c}^t\exp(2\Xi\mathfrak{a}(s-t))ds$$

(taking into account that Ξ and $\mathfrak{a}$ are symmetric). Replacing once again $\mathfrak{c}\mathfrak{c}^t$ by its average we obtain

$$\begin{aligned}
\operatorname{Cov}(\theta(t)) &= r\sigma^2\int_0^t \exp(2\mathfrak{a}\Xi(s-t))\mathfrak{a}\exp(2\Xi\mathfrak{a}(s-t))ds \\
&= r\sigma^2\mathfrak{a}\int_0^t \exp(4\Xi\mathfrak{a}(s-t))ds \\
&= \frac{1}{4}r\sigma^2\Xi^{-1}\exp(4\Xi\mathfrak{a}(s-t))|_0^t \\
&= \frac{1}{4}r\sigma^2\Xi^{-1}(1-\exp(-4\Xi\mathfrak{a}t))
\end{aligned}$$

Hence asymptotically

$$\mathrm{Cov}(\theta) = \frac{1}{4} r\sigma^2 \Xi^{-1}$$

and so asymptotically the distribution of θ is

$$\theta \sim N(0, \frac{1}{4} r\sigma^2 \Xi^{-1})$$

Recall the following.

Lemma 2.2. *Let Σ be a positive definite symmetric $n \times n$ matrix and*

$$X \sim N(0, \Sigma)$$

Then

$$X^t \Sigma^{-1} X \sim \chi^2_n$$

Proof. We may write $\Sigma = PP^t$. Put $Y = P^{-1}X$. Then $\mathrm{Cov}(Y) = P^{-1}\,\mathrm{Cov}(X)P^{-t} = P^{-1}\Sigma P^{-t} = I$. Hence $Y \sim N(0, I)$ and thus $Y^t Y \sim \chi^2_n$. But $Y^t Y = X^t P^{-t} P^{-1} X = X^t \Sigma^{-1} X$. □

Hence with this lemma we obtain asymptotically

$$\frac{4}{r\sigma^2} \theta^t \Xi \theta = -\frac{8f(\theta)}{r\sigma^2} \sim \chi^2_n$$

So

$$-\frac{8E(f(\theta))}{r\sigma^2} = n$$

from which we deduce (1), using the conversion (2.5) and (2.4). Also asymptotically

$$P\left(-\frac{8f(\theta)}{r\sigma^2} \leq z\right) = \gamma$$

Or

$$P\left(f(\theta) \geq -\frac{zr\sigma^2}{8}\right) = \gamma$$

from which we deduce (2), again using the conversion (2.5) and (2.4).

2.3. **The case of diagonal Hessian.** In this section we assume

$$e(\theta) = -\sum_i e_i \theta_i^2$$

and furthermore that we are in the situation of Example 1.2) so that

$$E(\mathfrak{c}\mathfrak{c}^{\mathfrak{t}}) = (c_1^2, \ldots, c_n^2)$$

Then Statement (3) maybe refined as follows.

(4) The expectation value of $\theta_i^{(j)}$, $j = 1, \ldots, n$ will converge linearly towards $\bar{\theta}^{(j)}$ as

$$\sim e^{-i/\lambda_j}$$

with the time constant λ_j given by

$$\lambda_j = \frac{C}{4re_i c_j^2} \tag{2.6}$$

We will now discuss some more detailed results. We have that $\theta_i(t)$ is normally distributed, and moreover

$$E(\theta_i(t)) = \exp(-4rc_i^2 e_i t/C)\theta_i(0)$$
$$\mathrm{Var}(\theta_i(t) = \frac{r(1-d)C}{8e_i}(1-\exp(-8e_i r c_i^2 t/C))$$

In particular we obtain

$$E(e(\theta(t))) = -\sum_i \left(e_i \exp(-8rc_i^2 e_i t/C)\theta_i^2(0) + \frac{r(1-d)C}{8}(1-\exp(-8rc_i^2 e_i t/C)) \right)$$

Furthermore $e(\theta(t))$ follows a *generalized χ^2-distribution* [1]. The SPSA simulator [5] contains routines to calculate the cumulative density function (CDF) and the percent point function (PPF) for the generalized χ^2-distribution. The CDF routine was originally based on MATLAB code written by Abhranil Das [2] and is based on Ruben's algorithm [4]. Essentially in Ruben's algorithm one develops the characteristic function $f(t)$ of a generalized χ^2-distribution as a power series in $(1-2itp)^{-1/2}$ for a suitable chosen p. This leads to a description of the generalized χ^2-distribution as a mixture of ordinary scaled χ^2_{n+2i}-distributions for $i \geq 0$.

It follows from Ruben's formulas that if $\exp(-8rc_i^2 e_i t/C) = \exp(-2t/\lambda_i)$ is small then the distribution of $e(\theta(t))$ satisfies approximately

$$\frac{n\theta(t)}{E(e(\theta(t)))} \sim \chi^2_n$$

We have also shown that one has *exactly*

$$\frac{n\theta(\infty)}{E(e(\theta(\infty)))} \sim \chi^2_n$$

It follows that

$$\frac{E(e(\theta(t)))}{E(e(\theta(\infty)))}$$

serves as a *scale factor* to translate asymptotic location quantities (e.g. mean and percentiles) into the corresponding quantities at time t for $t \gg 0$.

Let us finally note that if $c_i^2 e_i$ is independent of i so that we can put $\lambda = \lambda_j$ then we get the following elegant formula

$$E(\theta(t)) = E(\theta(0))\exp(-2t/\lambda) + E(\theta(\infty))(1-\exp(-2t/\lambda))$$

References

1. Wikipedia contributors, *Generalized chi-squared distribution*, `https://en.wikipedia.org/w/index.php?title=Generalized_chi-squared_distribution&oldid=958566492`.
2. A. Das, *MATLAB toolbox for classifying amongst normal distributions*, `https://github.com/abhranildas/classify`.
3. J. Kiiski, *SPSA Tuner for Stockfish Chess Engine*, `https://github.com/zamar/spsa`.
4. H. Ruben, *Probability content of regions under spherical normal distributions. IV. The distribution of homogeneous and non-homogeneous quadratic functions of normal variables*, Ann. Math. Statist. **33** (1962), 542–570.
5. M. Van den Bergh, *A multi threaded simulator for the chess version of the SPSA algorithm created by Joona Kiiski*, `https://github.com/vdbergh/spsa_simul`.
6. ———, *The accounting identiy*, `http://hardy.uhasselt.be/Fishtest/accounting_identity.pdf`.