A STOCHASTIC BEHAVIOR ANALYSIS OF STOCHASTIC RESTRICTED-GRADIENT DESCENT ALGORITHM IN REPRODUCING KERNEL HILBERT SPACES

Masa-aki Takizawa†, Masahiro Yukawa‡, and Cédric Richard‡

† Department of Electronics and Electrical Engineering, Keio University, Japan
‡ Université de Nice Sophia-Antipolis, CNRS, France

ABSTRACT

This paper presents a stochastic behavior analysis of a kernel-based stochastic restricted-gradient descent method. The restricted gradient gives a steepest ascent direction within the so-called dictionary subspace. The analysis provides the transient and steady state performance in the mean squared error criterion. It also includes stability conditions in the mean and mean-square sense. The present study is based on the analysis of the kernel normalized least mean square (KNLMS) algorithm initially proposed by Chen et al. Simulation results validate the analysis.

Index Terms— kernel adaptive filter, reproducing kernel Hilbert space, the KLMS algorithm, performance analysis

1. INTRODUCTION

Kernel adaptive filtering [1] is an attractive approach for nonlinear estimation problems based on the theory of reproducing kernel Hilbert space (RKHS), and a number of kernel adaptive filtering algorithms have been proposed [2–8]. The existing kernel adaptive filtering algorithms are classified into two general categories according to the space in which optimization is performed [6]: (i) the RKHS approach (e.g., [2, 5, 7]) and (ii) the parameter-space approach (e.g., [4, 6, 9]). The kernel normalized least mean square (KNLMS) algorithm is a representative example of the parameter-space approach and its stochastic behavior analyses have been presented in [10–12]. The analyses have clarified the transient and steady-state performance in the mean squared error (MSE). A stochastic restricted-gradient descent algorithm studied in the present work is an RKHS approach. Dictionary sparsification is a common issue of kernel adaptive filtering researches, let us give a short note on the HYPASS algorithm. We call it the natural kernel adaptive filter is given as a finite order filter:

\[ \varphi_n := \sum_{j \in J} \alpha_j^{(n)} \kappa(\cdot, u_j), \quad n \in \mathbb{N}, \]

where \( \alpha_j^{(n)} \in \mathbb{R} \) are the filter coefficients and \( J := \{ j_1, j_2, \ldots, j_r \} \) indicates the dictionary \( \{ \kappa(\cdot, u_j) \}_{j \in J} \). The instantaneous error at time instant \( n \) is defined as \( e_n := d_n - \langle \varphi, \kappa(\cdot, u_n) \rangle_\mathcal{H} - \langle \alpha, \kappa_n \rangle \), where \( \kappa_n = [\kappa(u_n, u_{j1}), \ldots, \kappa(u_n, u_{jr})] \).
\( \kappa(u_1, u_2, \cdots, u_n) \) is the vector of the kernelized input and \( \alpha = [\alpha_1, \alpha_2, \ldots, \alpha_r]^T \) is the coefficient vector. The MSE cost function, with respect to the coefficient vector \( \alpha \), is given by

\[
J(\alpha) := E(e_n^2(\alpha)) = E(d_n^2) + \alpha^T R_c \alpha - 2 \beta^T \alpha,
\]

where \( R_c := E(\kappa(u_n, \cdot)) \) is the autocorrelation matrix of the kernelized input \( \kappa_n \) and \( \beta := E(d_n, \kappa_n) \) is the cross-correlation vector between \( \kappa_n \) and \( d_n \). With the optimization in RKHS in mind, the MSE, with respect to \( \varphi \), is given by:

\[
J(\varphi) := E(e_n^2(\varphi)) = E(d_n^2) + \langle \varphi, \kappa(\cdot, u_n) \rangle^2 - 2E(d_n \langle \varphi, \kappa(\cdot, u_n) \rangle_R).
\]

While the KNLSM algorithm optimizes \( J(\alpha) \) in the Euclidean space \( \mathbb{R}^r \), the Natural KLMS algorithm presented in the following section optimizes \( J(\varphi) \) in the RKHS \( \mathcal{H} \) under the restriction to the dictionary subspace \( \mathcal{M} \), or in short, it optimizes \( J(\varphi) \) in \( \mathcal{M} \). Referring to [2], the stochastic gradient descent method for \( J(\varphi) \) in \( \mathcal{H} \) updates the filter \( \varphi_n \) along the ‘line’ (one dimensional subspace) spanned by the singleton \( \{\kappa(\cdot, u_n)\} \). This implies that the filter is updated only when \( \kappa(\cdot, u_n) \) is added into the dictionary, because otherwise \( \varphi_n + \alpha_n (\cdot, u_n) \notin \mathcal{M} \) for any \( \alpha_n \). We thus derive the restricted gradient, which was initially introduced in [17], and derive the Natural KLMS algorithm in the following section.

### 3. THE NATURAL KLMS ALGORITHM

The ordinary gradient of \( J(\alpha) \) in \( \mathbb{R}^r \) is given by \( \nabla J(\alpha) = 2(R_c \alpha - \beta) \). Given any positive definite matrix \( Q \), \( \langle x, y \rangle_Q := x^T Q y \) and \( \|x\|_Q := \sqrt{\langle x, x \rangle_Q} \) define an inner product and its induced norm, respectively. The \( G \)-gradient of (3) with the inner product \( \langle \cdot, \cdot \rangle_G \) is defined as [17]

\[
\nabla_G J(\alpha) := G^{-1} \nabla J(\alpha),
\]

where \( [G]_{\ell, m} = \kappa(u_\ell, u_m) \) for \( 1 \leq \ell, m \leq r \) is the Gram matrix.

The functional Hilbert space \( \{\kappa(\cdot, u_n)\}_n \) is isomorphic to the Hilbert space \( (\mathbb{R}^r, \langle \cdot, \cdot \rangle_G) \) under the correspondence (see Fig. 1)

\[
\mathcal{M} \ni \varphi := \sum_{j \in \mathcal{J}} \alpha_j \kappa(\cdot, u_j) \leftrightarrow \alpha := [\alpha_1, \ldots, \alpha_r]^T \in \mathbb{R}^r.
\]

Note here that the isomorphism as Hilbert spaces includes, in addition to the one-to-one correspondence between the elements, the preservation of the inner product; i.e., \( \langle \varphi_1, \varphi_2 \rangle_G = \langle \alpha_1, \alpha_2 \rangle_G \) for any \( \varphi_1 \leftrightarrow \alpha_1 \) and \( \varphi_2 \leftrightarrow \alpha_2 \). Under the correspondence in (6), the restricted gradient \( \nabla_{\mathcal{M}} J(\varphi) \) is defined, through the \( G \)-gradient in \( \mathbb{R}^r \), as follows [17]:

\[
\nabla_{\mathcal{M}} J(\varphi) \leftrightarrow \nabla_G J(\alpha) = G^{-1} \nabla J(\alpha).
\]

The restricted gradient \( \nabla_{\mathcal{M}} J(\varphi) \) gives the steepest ascent direction, within the dictionary subspace \( \mathcal{M} \), of the tangent plane of the functional (4) at the point \( \varphi \). See the derivation of the restricted gradient in [17]. An instantaneous approximation of the restricted gradient \( \nabla_{\mathcal{M}} J(\varphi_n) \leftrightarrow \nabla_G J(\alpha_n) \), where \( \alpha_n := [\alpha_1, \alpha_2, \ldots, \alpha_r]^T \in \mathbb{R}^r \), is given by (7).

\[ \alpha_n := \frac{\alpha_n}{\| \alpha_n \|_Q} \begin{pmatrix} \tilde{R}_n \alpha_n \end{pmatrix} \begin{pmatrix} \tilde{v}_n \end{pmatrix}, \]

where \( \tilde{R}_n = G^{-1} R_c G^{-1} \) and \( \tilde{v}_n = G^{-1} v_n \).

For any positive semi-definite matrix \( Q \), there exists a unique square root \( Q^\frac{1}{2} \) satisfying \( Q = Q^\frac{1}{2} Q^\frac{1}{2} \).
4.2. Mean weight error analysis

The estimation error can be expressed by

\[ e_n = d_n - \kappa_n^T \tilde{v}_n - \kappa_n^T \tilde{\alpha}. \]  

(15)

Substituting (15) to (9), we obtain the recursive expression for \( \tilde{v}_n \):

\[ \tilde{v}_{n+1} = \tilde{v}_n + \eta d_n \kappa_n - \eta \kappa_n^T \tilde{v}_n \kappa_n - \eta \kappa_n^T \tilde{\alpha} \kappa_n. \]  

(16)

Using CMIA, we obtain the mean weight error model

\[ E(\tilde{v}_{n+1}) = (I_r - \eta \tilde{R}_n) E(\tilde{v}_n), \]  

(17)

where \( I_r \) denotes the \( r \times r \) identity matrix for any positive integer \( r \). Let the input \( u_n \) be a random vector following a Gaussian distribution with zero mean and the covariance matrix \( R_u := E(u_n u_n^T) \). Then, the \((\ell, m)\) component \((1 \leq \ell, m \leq r)\) of the autocorrelation matrix \( R_n \) of \( \kappa_n \) is given by \([12]\):

\[
\langle R_n \rangle_{\ell,m} = \left[I_L + \frac{2}{\sigma^2} R_u^{-1} \right]_{\ell,m} \exp \left[-\frac{1}{4\sigma^2} \left(2 \|u_{\ell m}\|^2 - \|u_{\ell m}\|^2 (I_L + \frac{2}{\sigma^2} R_u^{-1})^{-1}\right)\right],
\]

where \( u_{\ell m} = u_{\ell} u_m \), \( \|u_{\ell m}\|^2 = \|u_{\ell}\|^2 + \|u_m\|^2 \), and \( \| \cdot \| \) stands for determinant.

From the recursion in (17), we obtain the mean stability condition of the Natural KLMS algorithm as follows.

**Theorem 1 (Stability in the mean)** Assume CMIA holds. Then, for any initial condition, given dictionary \( \{ \kappa(\cdot, u_n) \}_{n \in J} \), the Natural KLMS algorithm asymptotically converges in the mean if the step size is chosen to satisfy

\[ 0 < \eta < \frac{2}{\lambda_{\text{max}}(R_n)}, \]  

(18)

where \( \lambda_{\text{max}}(\cdot) \) denotes the maximum eigenvalue of the matrix.

**Proof:** It is clear from the well-known mean-stability results (see, e.g., [19]).

4.3. Mean-square error analysis

Squaring (15) and taking its expectation under CMIA, the MSE (10) of Natural KLMS can be rewritten as

\[ J(\tilde{\alpha}_n) = J_{\text{min}} + tr(\tilde{R}_n C_n), \]  

(19)

where \( \tilde{C}_n := E(\tilde{v}_n \tilde{v}_n^T) \) is the correlation matrix of \( \tilde{v}_n \) and \( J_{\text{min}} := E(d_n^2 - \tilde{\alpha}^T \tilde{\alpha}) \) is the minimum MSE. We assume \( \varepsilon_n := d_n - \tilde{\alpha}^T \tilde{\alpha} \) is sufficiently close to the optimal solution of the infinite order model so that \( E(\varepsilon_n^2) \approx 0 \), and \( \varepsilon_n \) and \( \kappa_n \) are uncorrelated. Following the arguments in [10, Section III, D] with \( \kappa_n \) and \( v \) replaced respectively by \( \kappa \) and \( \tilde{\alpha} \), we arrive, with simple manipulations, at the following recursion:

\[ \tilde{C}_{n+1} \approx \tilde{C}_n + \eta^2 (\tilde{T}_n + J_{\text{min}} \tilde{R}_n) - \eta (\tilde{R}_n \tilde{C}_n + \tilde{C}_n \tilde{R}_n), \]  

(20)

where \( \tilde{T}_n := E(\kappa_n \kappa_n^T \tilde{v}_n \tilde{v}_n^T \kappa_n \kappa_n^T) \) and its \((\ell, m)\) component can be approximated as

\[ \left[ \tilde{T}_n \right]_{\ell,m} \approx \operatorname{tr}(\tilde{S}_{\ell,m} \tilde{C}_n), \quad 1 \leq \ell, m \leq r. \]  

(21)

Here, the \((p,q)\) component \((1 \leq p, q \leq r)\) of \( \tilde{S}_{\ell,m} \) is defined as

\[ \left[ \tilde{S}_{\ell,m} \right]_{p,q} := E(\kappa_{\ell m} \kappa_p \kappa_{n,m} \kappa_q) = g_{\ell}^T H_{m,p} g_q, \]  

(22)

where \( \kappa_{\ell m} := \kappa_\ell \kappa_m \), \( g_{\ell} \) \((1 \leq \ell \leq r)\) is the \( \ell \)-th column vector of \( G^{-\frac{1}{2}} \), and \( H_{m,p} := E(\kappa_n \kappa_p g_m g_q) \). The approximation in (21) can be developed by following the arguments in [12, Section 3.3] with \( \kappa_n \) and \( v_n \) replaced by \( \kappa \) and \( \tilde{\alpha} \), respectively. Finally, the \((i,j)\) component of \( H_{m,p} \) can be written as

\[ [H_{m,p}]_{i,j} = g_m^T S_{i,j} g_p, \quad 1 \leq i, j \leq r, \]  

(23)

where \( [S_{i,j}]_{k,t} := E(\kappa_n \kappa_k \kappa_p \kappa_t) \), \( 1 \leq k, t \leq r \), with \( \kappa_{n,1} := \kappa(u_n, u_{n,i}) \) can be computed by \([12, Eq. (35)]\).

Let us now establish the mean-square stability condition and derive the steady-state MSE. Due to the presence of \( R_u \tilde{C}_n + \tilde{C}_n R_u \) in (20), we exploit the lexicographic representation of \( \tilde{C}_n \), i.e., the columns of each matrix are stacked on top of each other into a vector. The recursion (20) can be rewritten as

\[ \tilde{c}_{n+1} = K \tilde{c}_n + \eta^2 J_{\text{min}} \tilde{r}_n, \]  

(24)

where \( \tilde{c}_n \) and \( \tilde{r}_n \) are the lexicographic forms of \( \tilde{C}_n \) and \( R_u \), respectively, and

\[ K := I_{i,2} - \eta(K_1 + K_2) + \eta^2 K_3, \]  

(25)

where \( K_1 := I_r \otimes \tilde{R}_n, K_2 := \tilde{R}_n \otimes I_r \), and \( K_3 = I_{r^2 \times r^2} \) is an \( r \times r \) and \( r \times r \) matrix entries are: \([K_3]_{(\ell+m-1)r+q,(\ell+q-1)r} := [S_{\ell,m}]_{p,q} \) with \( 1 \leq \ell, m, p, q \leq r \). Here, \( \otimes \) denotes the Kronecker product. By (24) and (25), we obtain the following results.

**Theorem 2 (Mean-square stability)** Assume CMIA holds. For any initial conditions and \( \eta \) satisfying (18), given a dictionary \( \{\kappa(\cdot, u_n)\}_{n \in J} \), the Natural KLMS algorithm with Gaussian kernel is mean-square stable, if the matrix \( K \) is stable (i.e., the spectral radius of \( K \) is less than one).

**Proof:** The algorithm is said to be mean-square stable if, and only if, the state vector remains bounded and tends to a steady-state value, regardless of the initial condition [19]. To complete the proof, it is sufficient to show that \( \|\tilde{c}_n\|_F \) remains bounded and tends to a steady-state value, where \( F \) is a diagonal positive definite matrix. This is verified by the fact that \( \tilde{c}_n \) is bounded and tends to a steady-state value if the matrix \( K \) is stable.

**Theorem 3 (MSE in the steady state)** Consider a sufficiently small step size \( \eta \), which ensures mean and mean-square stability. The steady-state MSE is given by (19) with the lexicographic representation of \( \tilde{C}_\infty \) given by

\[ \tilde{c}_\infty = \eta^2 J_{\text{min}}(I_{i,2} - K)^{-1} \tilde{r}_n, \]  

(26)

provided that \( I_{i,2} - K \) is invertible.

**Proof:** Letting \( \tilde{c}_{n+1} = \tilde{c}_n \) in (24) and rearranging the equation, we obtain (26).

We remark on Theorem 3 that the invertibility of \( I_{i,2} - K \) is actually ensured by the stability of \( K \).

5. SIMULATION RESULTS

We shall compare simulated learning curves and analytic models to validate the analysis. We conduct two experiments under the same settings as in [12]. In the first experiment, the input sequence is generated by

\[ u_n := \rho u_{n-1} + \sigma_n \sqrt{1 - \rho^2} \omega_n, \]  

(27)

where \( \omega_n \) is the noise following the i.i.d standard normal distribution. The nonlinear system is defined as follows:

\[
\begin{cases}
x_n := 0.5 u_n - 0.3 u_{n-1}
\end{cases}
\]

\[ d_n := x_n - 0.5 x_n^2 + 0.1 x_n^3 + \nu_n, \]  

(28)

\[
\begin{cases}
\end{cases}
\]
Table 1. Computational complexity of the Natural KLMS algorithm.

|                  | Selective update | Full update   |
|------------------|------------------|--------------|
| Complexity       | \((L + s_n + 1)r + O(s_n^2)\) | \((L + r + 2)r\) |

where \(\nu_n\) is an additive zero-mean Gaussian noise with the standard deviation \(\sigma_\nu = 0.05\). The input vector is \(u_n = [u_n, u_{n-1}]^T\). The step size, the standard deviation of the input, the input correlation parameter, the kernel parameter and the dictionary size are set to \(\eta = 0.075\), \(\sigma_u = 0.5\), \(\rho = 0.5\), \(\sigma = 0.7\) and \(r = 25\), respectively. The dictionary is \(r\) samples on a uniform grid defined on \([-1, 1] \times [-1, 1]\).

Fig. 2(a) depicts the results: the learning curves, the theoretical transient MSE curve, and the theoretical steady state MSE line are presented in blue, red, and green (dotted line), respectively. The simulated curve is obtained by averaging over 300 Monte-Carlo runs. The theoretical MSE is estimated by (19) with \(C_n\) recursively evaluated by (20). The steady state MSE is computed by Theorem 3. Although the input is correlated, the theoretical MSE presented in Table 1 summarizes the overall per-iteration complexity (the number of real multiplications) of the Natural KLMS algorithm with full update and selective update (see [15, 18]), and Fig. 4 illustrates the complexity as a function of the dictionary size \(r\) for \(L = 2\) and \(s_n = 1\); \(O(s_n^2)\) is counted simply as \(s_n^2\). Here, \(s_n = 1\) means that only one coefficient is updated at each iteration and hence the complexity is reduced drastically. Fig 2(b) and 3(b) depict the MSE learning curves of the Natural KLMS algorithm with full update and selective update for \(s_n = 1\). It can be seen that the Natural KLMS algorithm with the selective update exhibits a steady-state MSE comparable to the full-update case with drastically lower complexity.

6. CONCLUSION

This paper presented a stochastic behavior analysis of the Natural KLMS algorithm which is a stochastic restricted-gradient descent method. The analysis provided a transient and steady-state MSEs of the algorithm. We also derived stability conditions in the mean and mean-square sense. Simulation results showed that the theoretical MSE curves given by the analysis well meet the simulated MSE curves. The outcomes of this study will serve as a theoretical basis to compare the performances of KNLMS and Natural KLMS.
7. REFERENCES

[1] W. Liu, J. Príncipe, and S. Haykin, *Kernel Adaptive Filtering*. New Jersey: Wiley, 2010.

[2] J. Kivinen, A. J. Smola, and R. C. Williamson, “Online learning with kernels,” *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2165–2176, Aug. 2004.

[3] Y. Engel, S. Mannor, and R. Meir, “The kernel recursive least-squares algorithm,” *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2275–2285, Aug. 2004.

[4] C. Richard, J.-C. M. Bermudez, and P. Honeine, “Online prediction of time series data with kernels,” *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 1058–1067, Mar. 2009.

[5] K. Slavakis, S. Theodoridis, and I. Yamada, “Adaptive constrained learning in reproducing kernel Hilbert spaces: the robust beamforming case,” *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4744–4764, Dec. 2009.

[6] M. Yukawa, “Multikernel adaptive filtering,” *IEEE Trans. Signal Processing*, vol. 60, no. 9, pp. 4672–4682, Sep. 2012.

[7] B. Chen, S. Zhao, P. Zhu, and J. C. Príncipe, “Quantized kernel least mean square algorithm,” *IEEE Trans. Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 22–32, 2012.

[8] S. V. Vaerenbergh, M. Lazaro-Gradilla, and I. Santamaria, “Kernel recursive least-squares tracker for time-varying regression,” *IEEE Trans. Neural Network and Learning Systems*, vol. 23, no. 8, pp. 1313–1326, Aug 2012.

[9] W. Gao, J. Chen, C. Richard, and J. Huang, “Online dictionary learning for kernel LMS,” *IEEE Trans. Signal Process.*, vol. 62, no. 11, pp. 2765–2777, June 2014.

[10] W. D. Parreira, J.-C. M. Bermudez, C. Richard, and J. Y. Tourneri, “Stochastic behavior analysis of the Gaussian kernel least-mean-square algorithm,” *IEEE Trans. Signal Processing*, vol. 60, no. 5, pp. 2208–2222, May 2012.

[11] C. Richard and J.-C. M. Bermudez, “Closed-form conditions for convergence of the gaussian kernel-least-mean-square algorithm,” in *Proc. Asilomar*, Pacific Grove, CA, USA, Nov. 2012, pp. 1797–1801.

[12] J. Chen, W. Gao, C. Richard, and J.-C. M. Bermudez, “Convergence analysis of kernel LMS algorithm with pre-tuned dictionary,” in *Proc. IEEE ICASSP*, 2014, pp. 7243–7247.

[13] W. Liu, P. P. Pokharel, and J. C. Príncipe, “The kernel least-mean-square algorithm,” *IEEE Trans. Signal Process.*, vol. 56, no. 2, pp. 543–554, Feb. 2008.

[14] J. Platt, “A resource-allocating network for function interpolation,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 3, no. 2, pp. 213–225, 1991.

[15] M. Yukawa and R. Ishii, “An efficient kernel adaptive filtering algorithm using hyperplane projection along affine subspace,” in *Proc. EUSIPCO*, 2012, pp. 2183–2187.

[16] B. Chen, S. Zhao, P. Zhu, and J. C. Príncipe, “Mean square convergence analysis for kernel least mean square algorithm,” *Signal Processing*, vol. 92, pp. 2624–2632, 2012.

[17] M. Takizawa and M. Yukawa, “An efficient sparse kernel adaptive filtering algorithm based on isomorphism between functional subspace and euclidean space,” in *Proc. IEEE ICASSP*, 2014, pp. 4508–4512.

[18] ——, “Adaptive nonlinear estimation based on parallel projection along affine subspaces in reproducing kernel hilbert space,” *IEEE Trans. Signal Processing*, submitted for publication.

[19] A. H. Sayed, *Adaptive Filters*. John Wiley & Sons, 2008.

[20] H. Al-Duwaish, M. N. Karim, and V. Chandrasekar, “Use of multilayer feedforward neural networks in identification and control of wiener model,” in *Proc. IEEE Control Theory Appl.*, vol. 143, 1996, pp. 255–258.