

A High-Order Recursive Quadratic Learning Algorithm

Qi Zhu^{1,*}, Shaohua Tan², and Ying Qiao³

¹ University of Houston - Victoria, TX 77901
Tel.: +1-361-570-4312, Fax: +1-361-570-4207
zhuq@uhv.edu

² National University of Singapore,
Singapore, 119260

³ Virginia Polytechnic Institute and State University,
Virginia, 24060

Abstract. A k -order Recursive Quadratic learning algorithm is proposed and its features are described in detail in this paper. Simulations are carried out to illustrate the efficiency and effectiveness of this new algorithm by comparing the results with both the projection algorithm and the conventional least squares algorithm.

1 Introduction

The Least Squares (LS) algorithm [3] and its adaptive version Recursive Least Squares (RLS) algorithm [2] are well-known algorithms widely used in areas such as Data Compression [8], Neural Network [5], Parameter identification [10], Pattern Recognition [4], Graphics [7], and Gene/DNA studies[9]. RLS is often used in linear-in-the-parameter (LIP) models. However, some undesirable features of RLS algorithm are that the regression matrix should be of full rank, convergence slows down considerably at certain low level error, and it persists over a considerable number of iteration steps before dropping eventually below the prescribed error bound. This is due to the fact that at low error level, RLS algorithm takes very small step sizes to ensure the convergence.

This paper develops a *high-order Recursive Quadratic (RQ for short) learning algorithm*, initially proposed in [11], which avoids the problem of RLS for identifying a general class of linear and nonlinear LIP models. RQ algorithm is thoroughly investigated and we reveal its features as a high-order extension of the Projection algorithm with a quadratic cost function. The convergence and accuracy analysis of the algorithm is performed along with the simulations to demonstrate various specialized properties of the algorithm.

* Corresponding author.

2 The RQ Algorithm

2.1 Preliminaries

The *Linear-in-the-Parameters (LIP)* is one of the most widely used model structures with the following general form [12]:

$$y(t) = \sum_{l=1}^m \xi_l(x(t))\omega_l = \varphi^T(x(t))\theta^* \quad (1)$$

where $\{x(t), y(t)\}(t > 0)$ is a set of sample data, $\theta^* = [\omega_1, \dots, \omega_l, \dots]^T$ is the desired weight, and $\varphi(x(t)) = [\xi_1(x(t)), \dots, \xi_l(x(t)), \dots]^T$ is an m -vector of basis functions.

We define a kt -dimensional matrix Λ_t to be the system forgetting factor:

$$\Lambda_t = \begin{bmatrix} 0_{k(t-1)} & 0 \\ 0 & \Lambda(kt, k) \end{bmatrix} \quad (2)$$

where $\Lambda(kt, k)$ is a k -dimensional diagonal matrix $\Lambda(kt, k) = \text{diag}[\lambda_1, \dots, \lambda_k]$ and λ_i ($i = 1, 2, \dots, k$) are some positive real scalars. The constant k is the order of the algorithm.

2.2 RQ Learning Algorithm for LIP Models

For a particular k , we define the abbreviated notations of input matrix, output vector, and output error vector as follows:

$$\Phi_t = \Phi_t(kt, k) \triangleq [\varphi_t(x(k(t-1)+1)), \varphi_t(x(k(t-1)+2)), \dots, \varphi_t(x(k(t-1)+k))]^T \quad (3)$$

$$Y_t = Y_t(kt, k) \triangleq [y_t(k(t-1)+1), y_t(k(t-1)+2), \dots, y_t(k(t-1)+k)]^T \quad (4)$$

$$\begin{aligned} E_t = E_t(kt, k) &\triangleq Y_t(kt, k) - \Phi_t(kt, k)\hat{\theta}_{t-1} \\ &= [e_t(k(t-1)+1), e_t(k(t-1)+2), \dots, e_t(k(t-1)+k)]^T \end{aligned} \quad (5)$$

where $e_t(kt) = y_t(kt) - \varphi_t^T(x(kt))\hat{\theta}_{t-1}$, subscript t denotes that the parameters are estimated at time t . Introduce J_t to be the quadratic function:

$$\begin{aligned} J_t = J_t(kt, k) &\triangleq \frac{1}{2}E_t^T \Lambda(kt, k)E_t = \frac{1}{2}(Y_t - \Phi_t\hat{\theta}_{t-1})^T \Lambda(kt, k)(Y_t - \Phi_t\hat{\theta}_{t-1}) \\ &= \frac{1}{2}\hat{\theta}_{t-1}^T \Phi_t^T \Lambda(kt, k)\Phi_t\hat{\theta}_{t-1} - \hat{\theta}_{t-1}^T \Phi_t^T \Lambda(kt, k)Y_t + \frac{1}{2}Y_t^T \Lambda(kt, k)Y_t \\ &= \frac{1}{2}\hat{\theta}_{t-1}^T P_t \hat{\theta}_{t-1} - \hat{\theta}_{t-1}^T Q_t + \frac{1}{2}R_t \end{aligned} \quad (6)$$

where $\Lambda(kt, k)$ is a k -dimensional identity matrix if we select $\lambda_i = 1(\forall i = 1, 2, \dots, k)$, and

$$\begin{aligned} P_t &= P_t(kt, k) \triangleq \Phi_t^T(kt, k)\Lambda(kt, k)\Phi_t(kt, k) = \Phi_t^T \Phi_t \\ Q_t &= Q_t(kt, k) \triangleq \Phi_t^T(kt, k)\Lambda(kt, k)Y_t(kt, k) = \Phi_t^T Y_t \\ R_t &= R_t(kt, k) \triangleq Y_t^T(kt, k)\Lambda(kt, k)Y_t(kt, k) = Y_t^T Y_t \end{aligned}$$

Using the above notations we can introduce **the k -order RQ algorithm** as follows:

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \frac{\alpha J_t(Q_t - P_t \hat{\theta}_{t-1})}{\beta + (Q_t - P_t \hat{\theta}_{t-1})^T (Q_t - P_t \hat{\theta}_{t-1})} \quad (7)$$

where $t = 1, 2, \dots$; and $\beta > 0$, $0 < \alpha < 4$.

Theorem 1. The algorithm (7) is obtained by solving the following optimization problem: Given $\hat{\theta}_{t-1}$ and Y_t , determine $\hat{\theta}_t$ so that $J_{\hat{\theta}} = \frac{1}{2} \|\hat{\theta}_t - \hat{\theta}_{t-1}\|^2$ is minimized subject to

$$(Y_t - \Phi_t \hat{\theta}_{t-1})^T Y_t = (Y_t - \Phi_t \hat{\theta}_{t-1})^T \Phi_t^T \hat{\theta}_t \quad (8)$$

Proof. Introducing a Lagrange multiplier λ for the constraint (8), we have the augmented function as $J'_{\hat{\theta}} = \frac{1}{2} \|\hat{\theta}_t - \hat{\theta}_{t-1}\|^2 + 2\lambda [Y_t - \Phi_t \hat{\theta}_{t-1}]^T [Y_t - \Phi_t \hat{\theta}_t]$. The necessary conditions for an optimization are $\partial J'_{\hat{\theta}} / \partial \hat{\theta}_t = 0$ and $\partial J'_{\hat{\theta}} / \partial \lambda = 0$, which are

$$\hat{\theta}_t - \hat{\theta}_{t-1} - 2\lambda \Phi_t^T [Y_t - \Phi_t \hat{\theta}_{t-1}] = 0 \quad (9)$$

$$[Y_t - \Phi_t \hat{\theta}_{t-1}]^T [Y_t - \Phi_t \hat{\theta}_t] = 0 \quad (10)$$

From (9) we obtain $\hat{\theta}_t = \hat{\theta}_{t-1} + 2\lambda \Phi_t^T [Y_t - \Phi_t \hat{\theta}_{t-1}]$, substituting into (10) gives

$$\lambda = \frac{\frac{1}{2} [Y_t - \Phi_t \hat{\theta}_{t-1}]^T [Y_t - \Phi_t \hat{\theta}_{t-1}]}{[Y_t - \Phi_t \hat{\theta}_{t-1}]^T \Phi_t \Phi_t^T [Y_t - \Phi_t \hat{\theta}_{t-1}]} \quad (11)$$

And we have $\Phi_t^T (Y_t - \Phi_t \hat{\theta}_{t-1}) = \Phi_t^T E_t = Q_t - P_t \hat{\theta}_{t-1}$, then substituting this as well as (5),(6), and (11) into (9) gives

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \frac{2J_t(Q_t - P_t \hat{\theta}_{t-1})}{(Q_t - P_t \hat{\theta}_{t-1})^T (Q_t - P_t \hat{\theta}_{t-1})}$$

To avoid division by zero, a small constant β is added to the denominator of the above formula. In order to adjust the convergence rate of the algorithm, we multiply a constant α to the numerator of the algorithm that erases the constant 2. This leads to the slightly modified form (7) of the new high-order RQ algorithm. \square

Figure 1 illustrates the geometric interpretation of the RQ algorithm with parameter of two dimensions $\hat{\theta} = (\theta_0 \ \theta_1)$ and the order $k = 4$. The parameters θ_0 and θ_1 span a plane if they are linearly independent. The input matrix $\Phi_t = \{\varphi_{t1}, \varphi_{t2}, \varphi_{t3}, \varphi_{t4}\}$ and each vector φ has two dimensions. Through Figure 1, we know that $\hat{\theta}_t$ is convergent to the desired parameter value θ^* using the shortest path, and it eventually reaches θ^* , i.e., $\|\hat{\theta}_t - \hat{\theta}_{t-1}\|$ is minimized by RQ learning algorithm.

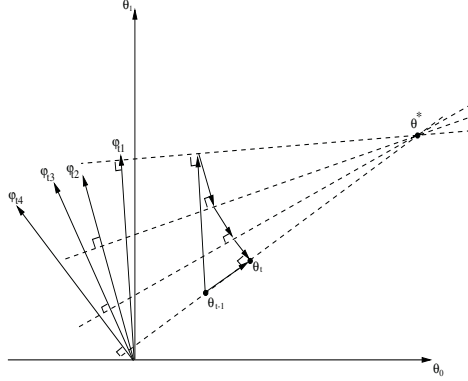


Fig. 1. Geometric interpretation of the Recursive Quadratic learning algorithm

2.3 Properties of RQ Learning Algorithm

Lemma 1 provides the convergent assessment of the RQ algorithm.

Lemma 1. For any given initial value $\hat{\theta}_0$, (7) has the following properties:

(i)

$$\|\hat{\theta}_t - \theta^*\| \leq \|\hat{\theta}_{t-1} - \theta^*\| \leq \dots \leq \|\hat{\theta}_0 - \theta^*\| \text{ for any } t \geq 1.$$

(ii)

$$\lim_{t \rightarrow \infty} J_t = 0$$

(iii)

$$\lim_{t \rightarrow \infty} \|\hat{\theta}_t - \hat{\theta}_{t-s}\| = 0 \text{ for any finite positive integer } s.$$

Proof. The complete proof is given in [11].

In Lemma 2, we consider the k -order RQ algorithm in the presence of a Gaussian noise.

Lemma 2. Modify the system model (1) into :

$$y_t = \varphi_t^T \theta^* + w_t \quad (12)$$

w_t is a sequence of Gaussian noise, so that $\lim_{t \rightarrow \infty} E[w_t] = 0$ and $\lim_{t \rightarrow \infty} E[w_t^2] = \sigma^2$. Define the k -order noise vector $W_t \triangleq [w_{k(t-1)+1}, w_{k(t-1)+2}, \dots, w_{k(t-1)+k}]^T$, then the RQ algorithm in (7) has the following properties:

(i) The output error converges in the mean

$$\lim_{t \rightarrow \infty} E[e_t] = 0$$

(ii)

$$\lim_{t \rightarrow \infty} E[e_t^T e_t] = \sigma^2$$

Proof. (i)

$$e_t = y_t - \varphi_t^T \hat{\theta}_{t-1} = -\varphi_t^T \tilde{\theta}_{t-1} + w_t$$

Then

$$\begin{aligned} \lim_{t \rightarrow \infty} E[e_t] &= \lim_{t \rightarrow \infty} E[-\varphi_t^T \tilde{\theta}_{t-1} + w_t] \\ &= -\lim_{t \rightarrow \infty} E[\varphi_t] \lim_{t \rightarrow \infty} E[\tilde{\theta}_{t-1}] + \lim_{t \rightarrow \infty} E[w_t] \end{aligned}$$

As w_t is white noise, $\lim_{t \rightarrow \infty} E[w_t] = 0$, with the parameter $\hat{\theta}_t$ being unbiased and the parameter error converging to zero, i.e. $\lim_{t \rightarrow \infty} \tilde{\theta}_{t-1} = \lim_{t \rightarrow \infty} (\hat{\theta}_{t-1} - \theta^*) = 0$, then $\lim_{t \rightarrow \infty} E[e_t] = 0$.

To prove (ii), we observe first that

$$\begin{aligned} \sum_{i=k(t-1)+1}^{kt} e_i^2 &= E_t^T E_t = (Y_t - \Phi_t \hat{\theta}_{t-1})^T (Y_t - \Phi_t \hat{\theta}_{t-1}) \\ &= (-\Phi_t \tilde{\theta}_{t-1} + W_t)^T (-\Phi_t \tilde{\theta}_{t-1} + W_t) \\ &= \tilde{\theta}_{t-1}^T \Phi_t^T \Phi_t \tilde{\theta}_{t-1} - \tilde{\theta}_{t-1}^T \Phi_t^T W_t - W_t^T \Phi_t \tilde{\theta}_{t-1} + W_t^T W_t \end{aligned}$$

The covariance estimate of the output error is:

$$\begin{aligned} \lim_{t \rightarrow \infty} E[E_t^T E_t] &= \lim_{t \rightarrow \infty} E[\tilde{\theta}_{t-1}^T \Phi_t^T \Phi_t \tilde{\theta}_{t-1}] - \lim_{t \rightarrow \infty} E[\tilde{\theta}_{t-1}^T \Phi_t^T] \lim_{t \rightarrow \infty} E[W_t] \\ &\quad - \lim_{t \rightarrow \infty} E[W_t^T] \lim_{t \rightarrow \infty} E[\Phi_t \tilde{\theta}_{t-1}] + \lim_{t \rightarrow \infty} E[W_t^T W_t] \end{aligned}$$

As the vector W_t is composed of a sequence of white noise w_t , we can conclude that $\lim_{t \rightarrow \infty} E[W_t] = 0_k$ is a k -order zero vector, and $\lim_{t \rightarrow \infty} E[W_t^T W_t] = k\sigma^2$. Since $\lim_{t \rightarrow \infty} \tilde{\theta}_{t-1} = 0$, and $\lim_{t \rightarrow \infty} \{\tilde{\theta}_{t-1}^T \Phi_t^T \Phi_t \tilde{\theta}_{t-1}\}$ is a scalar, we have

$$\begin{aligned} \lim_{t \rightarrow \infty} E[\tilde{\theta}_{t-1}^T \Phi_t^T \Phi_t \tilde{\theta}_{t-1}] &= \lim_{t \rightarrow \infty} E[\text{tr}\{\tilde{\theta}_{t-1}^T \Phi_t^T \Phi_t \tilde{\theta}_{t-1}\}] \\ &= \lim_{t \rightarrow \infty} E[\text{tr}\{\tilde{\theta}_{t-1}^T \tilde{\theta}_{t-1} \Phi_t^T \Phi_t\}] \\ &= \lim_{t \rightarrow \infty} \text{tr}\{E[\tilde{\theta}_{t-1}^T \tilde{\theta}_{t-1}] E[\Phi_t^T \Phi_t]\} = 0 \end{aligned}$$

Therefore

$$\lim_{t \rightarrow \infty} E[E_t^T E_t] = \lim_{t \rightarrow \infty} \sum_{i=k(t-1)+1}^{kt} E[e_i^2] = k\sigma^2$$

Finally

$$\lim_{t \rightarrow \infty} E[e_t^2] = \sigma^2$$

Lemma 2 allows us to conclude that (7) converges under the white noisy data as well. \square

Lemma 3. When $k = 1$, the new 1st-order RQ learning algorithm is equivalent to the Projection algorithm in [1].

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \frac{\alpha\phi_t}{\beta + \phi_t^T\phi_t} [y_t - \phi_t^T\hat{\theta}_{t-1}] \quad (13)$$

Proof. When $k = 1$, the 1st-order RQ learning algorithm becomes:

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \frac{\alpha'e_t^2\varphi_t}{\beta + e_t^2\varphi_t^T\varphi_t} (y_t - \varphi_t^T\hat{\theta}_{t-1}) \quad (14)$$

where $t > 0, \beta > 0, 0 < \alpha' = \frac{1}{2}\alpha < 2$, and $e_t = y_t - \varphi_t^T\hat{\theta}_{t-1}$.

Since β can be chosen as any positive number for preventing the denominator to be zero, selecting $\beta' = \beta/e_t^2$, (14) can be interpreted in terms of (13). Thus, the Projection algorithm is a special case of the new RQ learning algorithm when we choose a set of specific parameters. \square

3 Simulations

In this section, we present an example ¹ to assess the computational features of the RQ algorithm. We provide the performance comparisons among the k -order RQ learning algorithm, the projection algorithm, and the conventional Recursive Least Squares algorithm as well as the data statistics.

Example. The moving average (MA) model

$$\begin{aligned} x_t &= 0.1(t-1) \\ y_t &= 5 \sin x_t - 1 \cos x_t e^{\frac{1}{x_t+10}} + 2 \ln(x_t+10) + w_t \end{aligned}$$

where $t = 1, 6, 11, \dots, 101$ and w_t is a sequence of Gaussian noise with variance 0.01, $m = 3$ is the number of basis functions.

Figure 2 show the parameters convergence using 21 input and output data for identifying the system for $k = 15$, where k is the order of the RQ learning algorithm. With the choice of k , the number of multiplications (NM) is $3m^2 + 4m + km^2 + km + k + 3$ and the number of additions (NA) is $m^2 + m + m^2k + mk + k$.

These results are also compared to the Projection algorithm (13) and the RLS algorithm in [6]. For the Projection algorithm, the NM is $3m + 1$ and NA is $3m - 1$ per iteration step. For RLS algorithm, the NM is $6m^2 + 3m + 2$ and NA is $5m^2 - m$. We choose the same initial parameter value $\hat{\theta}_0 = [0, 0, 0]^T$ and the same error bound (3×10^{-4}) for all the three algorithms.

¹ Limit to space, we just show one example here. However, we have done many simulations for all kinds of LIP models

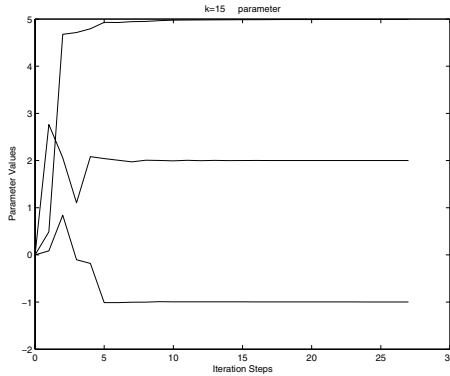


Fig. 2. When $k = 15$, RQ algorithm converges to 3×10^{-4} in 28 steps

The Projection algorithm can reach the error bound with 2339 iteration steps, its convergence rate is the slowest. The RLS algorithm converges faster than the Projection algorithm with 1387 iteration steps. However, after the initial fast convergence, the step length of the algorithm changes to be very small at low error level to avoid the convergence to the wrong parameter values.

The k -order RQ learning algorithm, on the other hand, can reach the error bound in only 28 iterations with $k = 15$. This shows that the speed of convergence of the k -order RQ learning algorithm is much faster than both the RLS algorithm and the Projection algorithm. Counting the total number of multiplications, additions and CPU time needed for the convergence, the RQ algorithm are 6636, 5796 and 1.43 seconds respectively, which are much less than RLS algorithm (90155 for NM, 58254 for NA and 22.57 seconds for CPU time) and Projection algorithm (23390 for NM, 18712 for NA and 71.70 seconds for CPU time).

We also observe that the choice of the order k is very critical. As shown in the Figure 3, if k is chosen to be large, then the convergence is fast but the

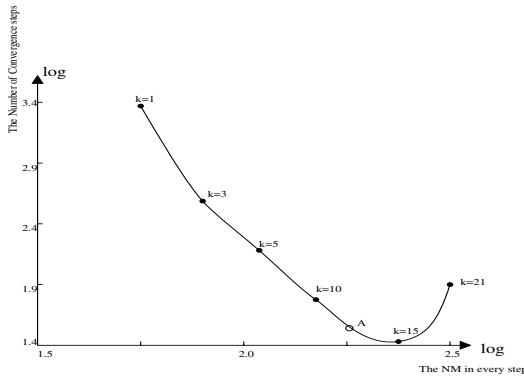


Fig. 3. Relationship between the steps and NM (both axes are scaled logarithmically)

computation at each iteration step is quite intensive. When k is too large, the convergence slows down again, at the same time, the NM and NA are very high. On the other hand, if k is chosen to be too small, then the computation is much simpler at each step, but it has much slower convergence. In this example, k around point A is the optimal choice, both the NM for each step and the total number of convergence steps are low. Currently the choice of k is based largely on intuitions rather than analytical means.

4 Conclusion

In this paper, we have developed a new high-order Recursive Quadratic (RQ) learning algorithm for Linear-in-the-Parameter models. This new RQ learning algorithm is derived as a high-order extension of the Projection algorithm with a new form of quadratic cost function. RQ algorithm is thoroughly described along with complete investigations to reveal its various features. The convergence and accuracy analysis of the algorithm is performed along with the simulations to demonstrate various specialized properties of the RQ algorithm.

We only developed the high-order RQ learning algorithm by choosing a specific A_t in this paper. One future research is to explore the learning algorithm by choosing different kinds of matrix A_t . In our research, choosing an appropriate order k is very critical. The larger the order, the faster the convergent speed and the more complex the computation at very iteration step. Hence, in future research it is very important that how can we choose a proper order k such that the convergent speed is reasonably fast, and yet the computation at every iteration step is reasonably simple in practice. Moreover, we can also extend the RQ learning algorithm for the purpose of on-line identification.

References

1. K.J. Astrom and B. Wittenmark, "Adaptive Control." Addison-Wesley, Mass., 1989.
2. Y. Boutalis, C. Papaodysseus, E. Koukoutsis, "A New Multichannel Recursive Least Squares Algorithm for Very Robust and Efficient Adaptive Filtering." *Journal of Algorithms*. Vol. 37, No.2, pp. 283-308, 2000.
3. L. E. Ghaoui and H. Le Bret, "Robust Least Squares and Applications." *In Proc. CDC*, pp. 249-254, Kobe, Japan, 1996.
4. S. Ghosal, R. Udupa, N. K. Ratha, S. Pankanti, "Hierarchical Partitioned Least Squares Filter-Bank for Fingerprint Enhancement." *In Proc. ICPR*. pp. 3338-3341, 2000.
5. A. P. Grinko, M. M. Karpuk, "Modification of Method of Least Squares for Tutoring Neural Networks." *In Proc. Intelligent Information Systems*. pp. 157-164, 2002.
6. T. C. Hsia, "System Identification: Least-Squares methods." Lexington Books, 1977.
7. H. A. Lensch, J. Kautz, M. Goesele, W. Heidrich, H. Seidel, "Image-based Reconstruction of Spatial Appearance and Geometric Detail." *ACM Transactions on Graphics*. Vol. 22, No. 2, pp: 234 - 257, 2003.

8. B. Meyer, P. E. Tischer, "Glicbawls - Grey Level Image Compression by Adaptive Weighted Least Squares." *Data Compression Conference*. pp. 503, 2001.
9. D. V. Nguyen, D. M. Rocke, "Multi-class Cancer Classification Via Partial Least Squares with Gene Expression Profiles. " *Bioinformatics*. Vol. 18, No.9, pp. 1216-1226, 2002.
10. B. D. Saunders, "Black Box Methods for Least Squares Problems." *In Proc. ISSAC*. pp. 297-302, 2001.
11. S. Tan, X. H. Zhang and Q. Zhu, "On a new k th-order Quadratic Learning Algorithm." *IEEE Trans. Circ. and Syst.* Vol. 44, No. 1, pp. 186-190, 1997 .
12. W. X. Zheng, "Modified Least-squares Identification of Linear Systems with Noisy Input and Output Observations." *In Proc. CDC*, pp. 1067-1068, Kobe, Japan, 1996.