Hindawi
Journal of Mathematics
Volume 2021, Article ID 7625175, 8 pages
https://doi.org/10.1155/2021/7625175

Research Article

An Efficient Algorithm for Ill-Conditioned Separable Nonlinear Least Squares

Jiayan Wang,[1,2] **Lanlan Guo**,[2] **Zongmin Li**,[1] **Xueqin Wang**,[2] **and Zhengqing Fu**[2]

[1]*University of Petroleum (Huadong), Qingdao, Shandong 266590, China*
[2]*Shandong University of Science and Technology, Qingdao, Shandong 266590, China*

Correspondence should be addressed to Zongmin Li; lizongmin@upc.edu.cn

Received 13 June 2021; Revised 8 September 2021; Accepted 18 October 2021; Published 10 November 2021

Academic Editor: Carmelo Antonio Finocchiaro

For separable nonlinear least squares models, a variable projection algorithm based on matrix factorization is studied, and the ill-conditioning of the model parameters is considered in the specific solution process of the model. When the linear parameters are estimated, the Tikhonov regularization method is used to solve the ill-conditioned problems. When the nonlinear parameters are estimated, the QR decomposition, Gram–Schmidt orthogonalization decomposition, and SVD are applied in the Jacobian matrix. These methods are then compared with the method in which the variables are not separated. Numerical experiments are performed using RBF neural network data, and the experimental results are analyzed in terms of both qualitative and quantitative indicators. The results show that the proposed algorithms are effective and robust.

1. Introduction

The nonlinear least squares model has been applied to solve several practical problems in many subject areas. The separable nonlinear least squares (SNLLS) problem is a special type of nonlinear least squares problem. The corresponding estimation model can be expressed as a linear combination of nonlinear problems. Based on the special structure of this model, Golub and Pereyra proposed a variable projection (VP) algorithm to solve the model in 1973 [1]. This algorithm can eliminate the linear parameters, transform the problem into an estimation problem with only nonlinear parameters, and reduce the complexity of the model. The VP algorithm can reduce the dimensions of the parameter space and the number of iterations, thus improving the efficiency of finding the global optimal solution to the SNLLS problem.

SNLLS models have a wide range of applications in engineering. Researchers have combined the VP algorithm with matrix decomposition or with rank recognition algorithms, which can be used to estimate different system models. Machine learning, neural networks and their many variants, and neural fuzzy systems [2] are all linear combinations of problems, where the problem is transformed into a nonlinear basis function. Prony converted a signal processing problem into a combination of complex exponential functions and used this method to analyze the frequency component of the signal. In time-series analyses, the smooth transition autoregressive model can be used as the separable least squares model. RBF-AR neural network models can be processed with linear combinations of exponential functions [3]. These examples are only part of the problem that can be solved using the SNLLS models.

In recent years, many improvements have been made to the VP algorithm. Krogh presented a more efficient VP algorithm. Kaufman introduced a simplified Jacobian matrix calculation method in the VP algorithm, which improved the calculation efficiency [4]. Ruhe and Wedin analyzed the asymptotic convergence of the VP algorithm [5]. Gan proposed a new VP formula that can solve optimization problems in the absence of a derivative [6]. Most forms of the VP method require matrix decomposition. O'Leary and Rust proposed a robust singular value decomposition (SVD) method [7]. Ruano et al. proposed an error function based on QR classification [8]. Gan [9] and Wang [10] proposed a classical and improved Gram–Schmidt orthogonalization (GSO) method.

Regarding the SNLLS problem, there are few studies on the potential ill-conditioning of the parameters to be estimated. Regularization is a commonly used method when ill-conditioned problems are solved. The use of regularization can make the regression coefficients have lower variance values, thereby reducing potential ill-conditioned problems [11, 12]. Commonly used regularization methods include the Tikhonov regularization method, the truncated singular value method, the kernel-based regularization method, and the norm-based regularization method [13–15].

In this study, the Levenberg–Marquardt (LM) algorithm is used to estimate the nonlinear parameters, and the Jacobian matrix in the algorithm is in the form given by Kaufman. During the iteration process, the SVD, QR decomposition, and GSO decomposition are applied to the matrix to improve the iteration efficiency. When the linear parameters are estimated, the Tikhonov regularization method is used to solve the potential ill-conditioned problems. Combining the advantages of the two methods improves not only the calculation efficiency of the algorithm but also the accuracy of the results. With RBF neural network data, the method is verified through both qualitative and quantitative analyses.

2. Variable Projection Model and Its Parameter Estimation Method

2.1. Variable Projection Model. The SNLLS model can be expressed as follows:

$$y(t) = \sum_{j=1}^{m} \theta_{L,j} \phi_j(\theta_N; t), \tag{1}$$

where $\phi_j(\theta_N; t)(j = 1, 2, \ldots, m, t = 1, 2, \ldots, n)$ are nonlinear functions. The linear parameters $\theta_L = (\theta_{L,1}, \theta_{L,2}, \ldots, \theta_{L,a})$ and nonlinear parameters $\theta_N = (\theta_{N,1}, \theta_{N,2}, \ldots, \theta_{N,b})$ are the parameters that need to be estimated.

The parameter estimators $\widehat{\theta}_L$ and $\widehat{\theta}_N$ can be obtained by solving the following formula:

$$\left(\widehat{\theta}_L, \widehat{\theta}_N\right) = \min_{\widehat{\theta}_L, \widehat{\theta}_N} \sum_{t=1}^{n} \left[y(t) - \sum_{j}^{m} \theta_L \phi_j(\theta_N; t) \right]^2. \tag{2}$$

The estimator can be expressed in a matrix form as

$$\left(\widehat{\theta}_L, \widehat{\theta}_N\right) = \min_{\widehat{\theta}_L, \widehat{\theta}_N} \left\| y - \Phi(\theta_N)\theta_L \right\|_2, \tag{3}$$

where $\Phi(\theta_N)$ is composed of a column vector $\phi_j(\theta_N; t)$. The component of the vector y is $y(t)$. $\|\cdot\|_2$ is the Euclidean norm of the vector.

We first estimate the nonlinear parameters and then estimate the linear parameters using the following nonlinear least squares method:

$$\widehat{\theta}_L = \left(\Phi(\theta_N)^T \Phi(\theta_N)\right)^{-1} \Phi(\theta_N)^T y = \Phi(\theta_N)^+ y. \tag{4}$$

In the above formula, $\Phi(\theta_N)^+$ is the pseudo-inverse of $\Phi(\theta_N)$.

In this paper, the separable nonlinear model can be expressed as follows:

$$y = \Phi(\theta_N)\theta_L = \Phi\theta_L, \tag{5}$$

where $\Phi \in R^{m \times a}$, $\theta_L \in R^{a \times 1}$, and $y \in R^{m \times 1}$.

The LM iterative algorithm is used to estimate the nonlinear parameters. In the calculation process, the calculation of the Jacobian matrix is important. For a matrix $F(\theta_N^k, \theta_L^k) = y - \Phi(\theta_N)\theta_L$, the Jacobian matrix is its first-derivative matrix.

$$\begin{aligned} J\left(\theta_N^k\right) &= \left(\frac{\partial F\left(\theta_N^k, \theta_L^k\right)}{\partial \theta_N^k} \right)^T, \\ P^{\perp}_{\Phi\left(\theta_N^k\right)} &= \left(I - \Phi\left(\theta_N^k\right) \Phi\left(\theta_N^k\right)^+ \right). \end{aligned} \tag{6}$$

Thus, we have

$$F\left(\theta_N^k, \theta_L^k\right) = \left(I - \Phi(\theta_N) \Phi\left(\theta_N^k\right)^+ \right) y. \tag{7}$$

In this model, the Jacobian matrix can be calculated using the method proposed by Kaufman [3].

$$J_{\mathrm{KAU}} = -P_{\Phi}^{\perp} D\Phi\ \Phi^- y, \tag{8}$$

where D is the Fréchet derivative of the map, Φ^- is the symmetric pseudo-inverse of Φ that satisfies $\Phi\Phi^-\Phi = \Phi$, and $(\Phi\Phi^-)^T = \Phi\Phi^-$.

$$r(\theta_N) = \left\| y - \Phi\left(\theta_N^k\right) \Phi\left(\theta_N^k\right)^+ y \right\|_2 = \left\| (I - P_{\Phi}) y \right\|_2 = \left\| P_{\Phi}^{\perp} y \right\|_2. \tag{9}$$

2.2. Nonlinear Parameter Estimation Method. To improve the calculation efficiency, the matrix is decomposed first, and then iterative calculations are performed with the decomposed matrix. The SVD, QR decomposition, and GSO decomposition are used.

2.2.1. SVD. The specific decomposition form is as follows:

$$\Phi = [U_1 U_2] \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} [V_1, V_2]^T, \tag{10}$$

where $\mathrm{rank}(U_1) = \mathrm{rank}(V_1) = \mathrm{rank}(\Phi) = r$ and Σ is a diagonal matrix of order r. Through SVD, we can obtain

$$\Phi^- = [V_1, V_2] \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{bmatrix} [U_1, U_2]^T. \tag{11}$$

The corresponding residual function and Jacobian matrix can be expressed as follows:

$$r_{\mathrm{SVD}}(\theta_N) = \left\|P_\Phi^\perp y\right\|_2 = \left\|[U_1U_2]\begin{bmatrix}0 & 0\\ 0 & I_{m-r}\end{bmatrix}[V_1,V_2]^T y\right\|_2$$
$$= \left\|U_2V_2 y\right\|_2$$
$$J_{\mathrm{SVD}} = -U_2U_2^T D\Phi\ V_1\Sigma^{-1}U_1^T y. \tag{12}$$

The algorithm based on the SVD is denoted by $\mathrm{VP}_{\mathrm{SVD}}$.

2.2.2. GSO Decomposition. The specific decomposition form is as follows:

$$\Phi = S\begin{bmatrix}T_{11} & T_{12}\\ 0 & 0\end{bmatrix}P^T = [S_1,S_2]\begin{bmatrix}T_{11} & T_{12}\\ 0 & 0\end{bmatrix}[P_1,P_2]^T, \tag{13}$$

where S is an orthogonal matrix, P is a permutation matrix, and T_{11} is an upper triangular matrix. Here, $\mathrm{rank}(s) = m, \mathrm{rank}(P) = n$, and $\mathrm{rank}(T_{11}) = r$. Through GSO decomposition, we can obtain

$$\Phi^- = [P_1P_2]\begin{bmatrix}T_{11}^{-1} & 0\\ 0 & 0\end{bmatrix}[S_1,S_2]^T. \tag{14}$$

The corresponding residual function and Jacobian matrix can be expressed as follows:

$$r_T(\theta_N)A = \left\|P_\Phi^\perp y\right\|_2 = \left\|S_2S_2^T y\right\|_2,$$
$$J_{\mathrm{GSO}} = -S_2S_2^T D\Phi\ P_1T_{11}^{-1}S_1^T y. \tag{15}$$

The algorithm based on the GSO decomposition is denoted by $\mathrm{VP}_{\mathrm{GSO}}$.

2.2.3. QR Decomposition. The specific decomposition form is as follows:

$$\Phi = \mathrm{QR} = [Q_1,Q_2]\begin{bmatrix}R_1\\ 0\end{bmatrix}, \tag{16}$$

where Q is an orthogonal matrix and R and R_1 are upper triangular matrices. The corresponding residual function and Jacobian matrix can be expressed as follows:

$$r_{\mathrm{QR}} = \left\|Q_2Q_2^T y\right\|_2,$$
$$J_{\mathrm{QR}} = -Q_2Q_2^T D\Phi\ R_1^{-1}Q_1 y. \tag{17}$$

The algorithm based on the QR decomposition is denoted by $\mathrm{VP}_{\mathrm{QR}}$.

The following iterative methods can be used to estimate the nonlinear parameters:

$$\theta_N^{k+1} = \theta_N^k + \beta_k d_k, \tag{18}$$

where β_k is the step size, which can make the objective function $\|y - \Phi(\theta_N)\widetilde{\theta}_L\|_2$ decrease.

d_k is the search direction, which can be determined by solving the following equation:

$$\left[J\left(\theta_N^k\right)^T J\left(\theta_N^k\right) + \gamma_k I\right]d_k = -J\left(\theta_N^k\right)^T F\left(\theta_N^k, \widetilde{\theta}_L^k\right), \tag{19}$$

where γ_k is the damping factor.

In the LM algorithm, β_k can be calculated using the formula

$$r\left(\theta^k + \rho^{m_k}d_k\right) \le r\left(\theta^k\right) + \rho^{m_k}g_k^T d_k. \tag{20}$$

m_k in (20) is the nonnegative minimum integer satisfying equation (20). Here, $\rho = 0.5$, $g_k = J(\theta_k)r(\theta_k)$, and $\beta_k = \rho^{m_k}$.

2.3. Linear Parameter Estimation Method. The SVD of matrix Φ can be expressed as

$$\Phi = \mathrm{USV}^T = \sum_{i=1}^{l} u_i\sigma_i v_i^T, \tag{21}$$

where U and V are the unitary matrices composed of column vectors.

$S = (s_{ij}) \in R^{a\times b}$, $s_{ij} = \begin{cases}\sigma_i\,(i=j)\\ 0\,(i\ne j)\end{cases}$, $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_l > 0$, $\sigma_{l+1} = \sigma_{l+2} = \cdots = \sigma_b = 0$, and $\{\sigma_i\}_{i=1}^l$ is the singular value of Φ. Solving equation (21) using the SVD method yields

$$\widehat{\theta}_L = \sum_{i=1}^{l}\frac{u_i^T y}{\sigma_i}v_i. \tag{22}$$

If the singular value decreases and reaches zero, then (21) may be an ill-conditioned problem. If the singular value cannot be zero, when σ_1/σ_l is very high and the condition number of Φ is also high, then (21) may be an ill-conditioned problem.

With the addition of filtering factors, the ill-conditioning is controlled to obtain ideal parameter estimates. After adding the filter factor, the SVD can be expressed as follows:

$$\widehat{\theta}_L = \sum_{i=1}^{l} f_i\frac{u_i^T y}{\sigma_i}v_i, \tag{23}$$

where the filter factor is $f_i = \sigma_i^2/\sigma_i^2 + \sigma_1^2$ and $\widehat{\theta}_L$ is obtained using the Tikhonov method. Equation (27) can be expressed as follows:

$$\widehat{\theta}_L = \sum_{i=1}^{l}\frac{\sigma_i^2}{\sigma_i^2+\sigma_1^2}\frac{u_i^T y}{\sigma_i}v_i. \tag{24}$$

By combining the linear parameter and nonlinear parameter estimation methods, three algorithms can be obtained. The LM algorithm is used to estimate the nonlinear parameters. During the iterative process, the Jacobian matrix takes three forms—J_{SVD}, J_{GSO}, and J_{QR}—for the SVD, GSO, and QR methods, respectively. The least squares method is used to estimate the linear parameters. During the iterative process, the Tikhonov regularization method is used to control the ill-conditioning. After the combination, the following estimation method can be obtained:

$$\left(\widehat{\theta}_L, \widehat{\theta}_N\right) = \arg\left\{ \min_{\widehat{\theta}_N} \left\| y - \Phi\left(\theta_N\right)\widehat{\theta}_L \right\|_2^2, \widehat{\theta}_L = \sum_{i=1}^{l} \frac{\sigma_i^2}{\sigma_i^2 + \sigma_1^2} \frac{u_i^T y}{\sigma_i} v_i \right\}. \tag{25}$$

The specific calculation process of the algorithm is as follows:

(1) Set the initial values of the nonlinear parameter θ_N^0, control error ε, and maximum number of iteration steps $k_{\max}$

(2) Use equation (21) to calculate the initial value of the linear parameter θ_L^0

(3) Calculate the linear and nonlinear parameters sequentially and iteratively

(4) Repeat the third step until it meets the condition $|\theta_N^{k+1} - \theta_N^k| < \varepsilon$ or reaches the maximum number of iterations $k_{\max}$

3. Numerical Experiments

The experiments were conducted on a PC with a 2.30 GHz processor and 4 GB of memory, running on Windows 10. Software used was MATLAB 2016b. The experiment was conducted to verify the three algorithms proposed in this paper.

In this experiment, the Mackey–Glass differential equation was used to generate a sequence of points.

$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} = \frac{a y(t-\tau)}{1 + y^c(t-\tau)} - b y(t), \tag{26}$$

where the values of the parameter are $a = 0.2$, $b = 0.1$, $c = 10$, and $\tau = 10$. The fourth-order Runge–Kutta method was used to intercept 500 points to test the SNLLS model. Figure 1 shows the specific point chart. The horizontal and vertical axes are the value points of x and y, respectively.

First, the first 200 points were used to learn and train the model and estimate the linear and nonlinear parameters. Subsequently, the remaining points were used to verify the identified model.

The RBF − AR(m, d) model is a linear combination of Gaussian RBFs, expressed as

$$\begin{aligned} y(t) &= \theta_{L,1} + \sum_{k=1}^{m} \theta_{L,k+1} \exp\left\{-\lambda_k \left\| t_t - z_k \right\|_2\right\} + \xi(t) \\ &= \sum_{j=1}^{m+1} \theta_{L,j} \phi_{j-1}\left(\theta_N; t_t\right) + \xi(t), \end{aligned} \tag{27}$$

where $m, d \in Z^+$ is the model number, λ_k is the model scaling parameters, z_k, $(k = 0, 1, \ldots, m)$ is the center of the model, θ_L represents the linear parameters, and θ_N represents the nonlinear parameters. The number of linear parameters in the RBF-AR model is often greater than the number of nonlinear parameters. The parameter estimation using the VP theory is suitable. The scaling parameter λ_k can be calculated using the following formula:

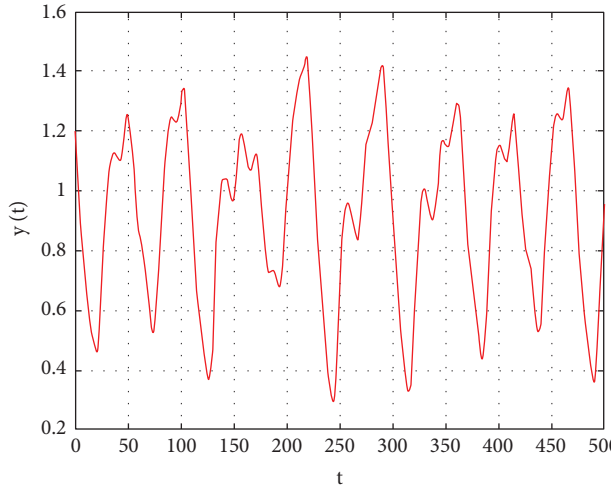

Figure 1: Mackey–Glass time series.

$$\lambda_k = -\log \frac{\varepsilon}{\max\left\{\left\| t_t - z_k \right\|_2\right\}}, 0.0001 \le \varepsilon \le 0.1. \tag{28}$$

When the model is solved, the following four methods are used: parameter nonseparation method, $\mathrm{VP}_{\mathrm{QR}}$, $\mathrm{VP}_{\mathrm{GSO}}$, and $\mathrm{VP}_{\mathrm{SVD}}$. Figure 2 shows the curves of the training and prediction points obtained using different methods. The first half of the curve represents the parameter training curve, containing 200 points. The parameters were estimated through these points, and a neural network model was established. The latter half of the curve represents the prediction curve obtained using the estimated model.

Figure 2 shows that the curves obtained using the four methods are consistent with the Mackey–Glass time-series curve. This shows that the four models are accurate and that they can predict the curve well.

The residual error and RMSE values of the four methods are low. The calculation formula of RMSE is shown in the following:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} \left(\theta_{N_o,i} - \theta_{N_e,i}\right)^2}{n}}, \tag{29}$$

where $\theta_{n_o,i}$ and $\theta_{n_e,i}$ are the observed value and estimated value.

Table 1 lists the quantitative indicators.

As listed in Table 1, this shows that they yield accurate estimation parameters and prediction curves. The residual and RMSE of the three proposed methods are lower than those of the method without matrix decomposition. The proposed methods have higher accuracy. Therefore, the three methods proposed in this article are compared in the following part.

Figure 3 shows the RMSE curves of the multistep prediction of the three methods. The horizontal axis is the number of prediction points. The vertical axis is the RMSE value.

As shown, with the increase in the number of prediction points, the three methods have a higher error growth rate in the early stages and a lower error growth rate in the later

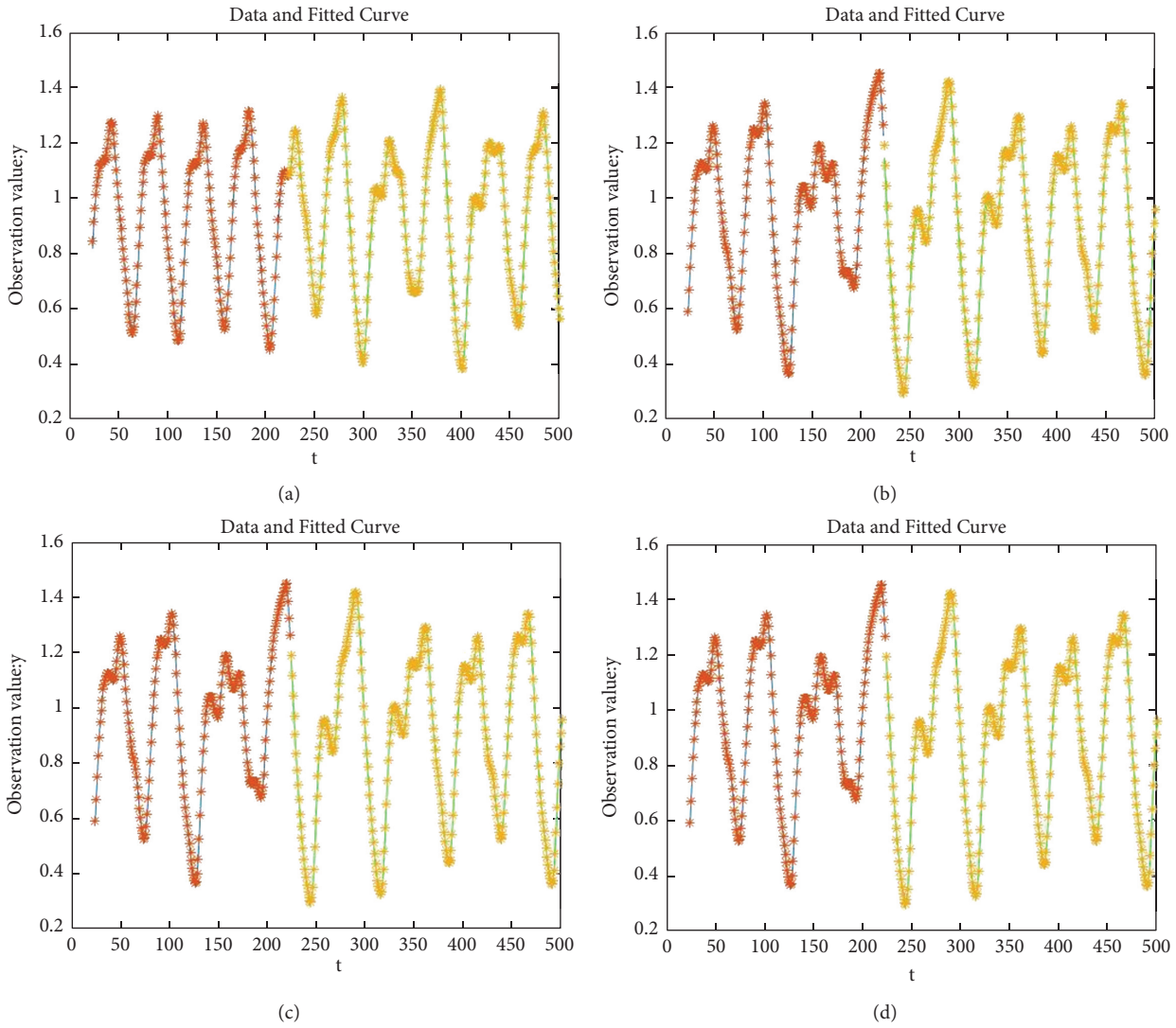

Figure 2: Curves of training and prediction points: (a) not separated, (b) QR, (c) GSO, and (d) SVD.

Table 1: Quantitative indicators.

	Not separated	VP_{QR}	VP_{SVD}	VP_{GSO}
Residuals of training points	0.1340	1.5349×10^{-8}	0.0260	0.0421
RMSE of training points	0.4177	6.1945×10^{-5}	0.0806	0.1025
Residuals of predicted points	0.2831	5.8095×10^{-8}	0.0936	0.3554
RMSE of predicted points	0.5949	1.2051×10^{-4}	0.1530	0.2981
Iteration time (s)	17368	719	542	528

stages. This shows that the proposed methods can accurately estimate the model parameters. The RMSE is the lowest for VP_{QR} and highest for VP_{SVD}; however, the difference is small, indicating the reliability of the parameters obtained by using the three methods.

With a random initial point, the experiment was repeated 30 times, and the box plots of the RMSE were obtained, as shown in Figure 4. The vertical axis is the RMSE value. From left to right are the box plots for the VP_{SVD}, VP_{GSO}, and VP_{QR} methods.

The residuals of the prediction points of the three methods are relatively stable. From the graph, it is found that the prediction point residuals obtained by using the VP_{QR} method are the most stable.

The LM algorithm was used to estimate the nonlinear parameters. Figure 5 shows the iterative convergence curves of the nonlinear parameters of the three methods.

From the iterative convergence of the nonlinear parameters, it is found that, after multiple iterations, the nonlinear parameters yield stable estimates. The three matrix

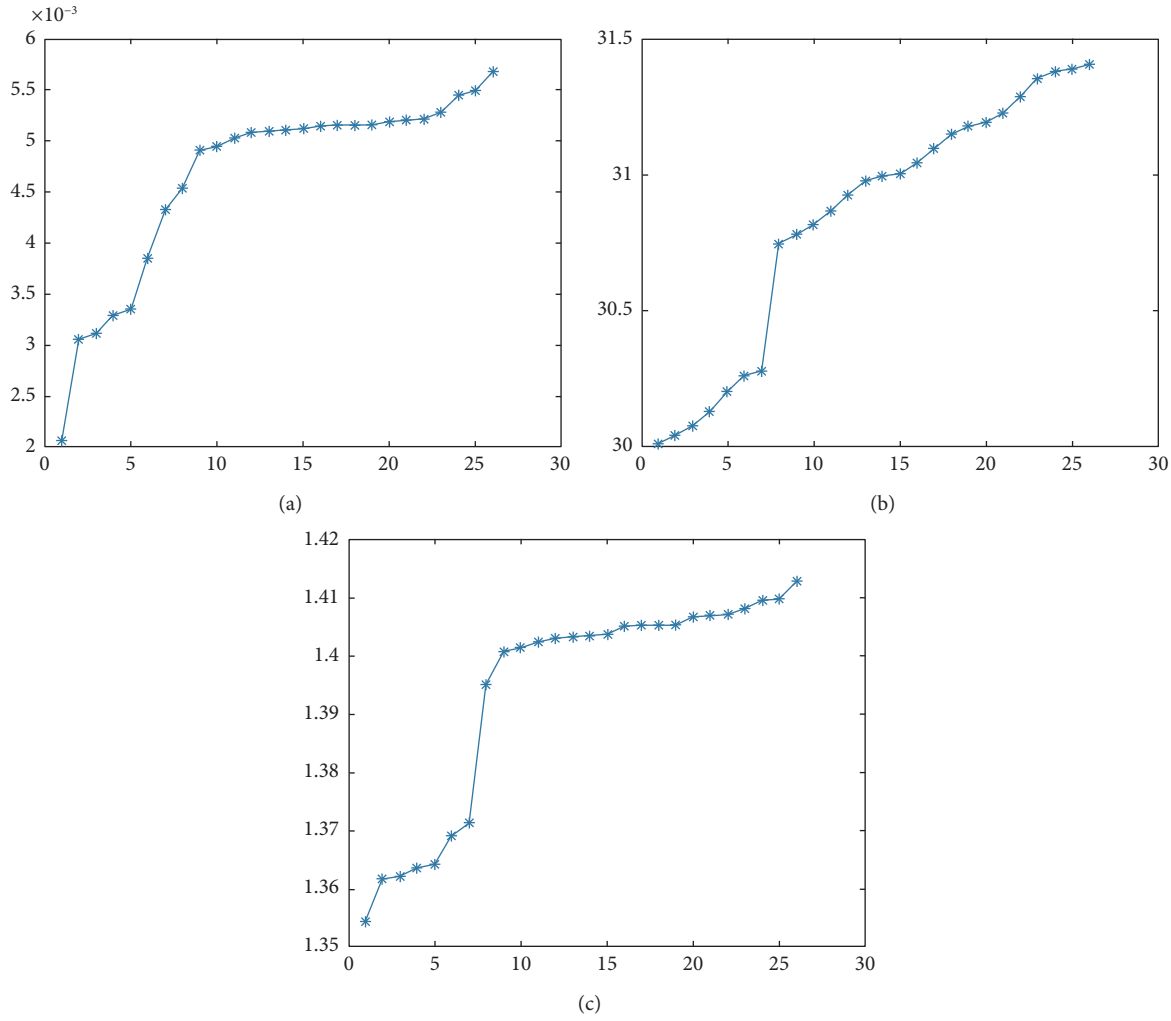

Figure 3: RMSE values of the multistep prediction of three methods: (a) QR, (b) GSO, and (c) SVD.

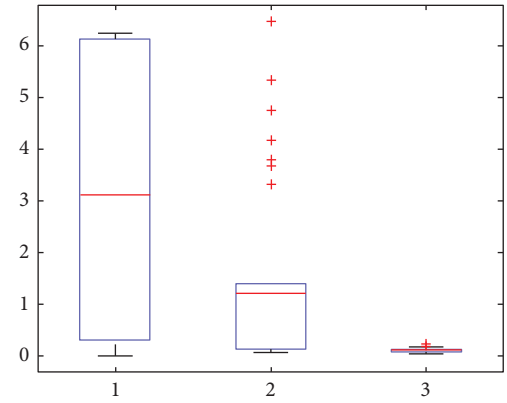

Figure 4: Box plots of the RMSE for three methods.

decompositions are all orthogonal decompositions. When solving the model, the three methods require different numbers of iterations. The VP_{GSO} method requires fewer iterations to reach a stable estimate. When using the LM algorithm to estimate the nonlinear parameters, the total residual change after each iteration of the parameter is calculated. The rise or fall of the curve during the iteration process does not mean that the algorithm is unstable. All the three parameter estimation methods stably converge to the optimal estimate.

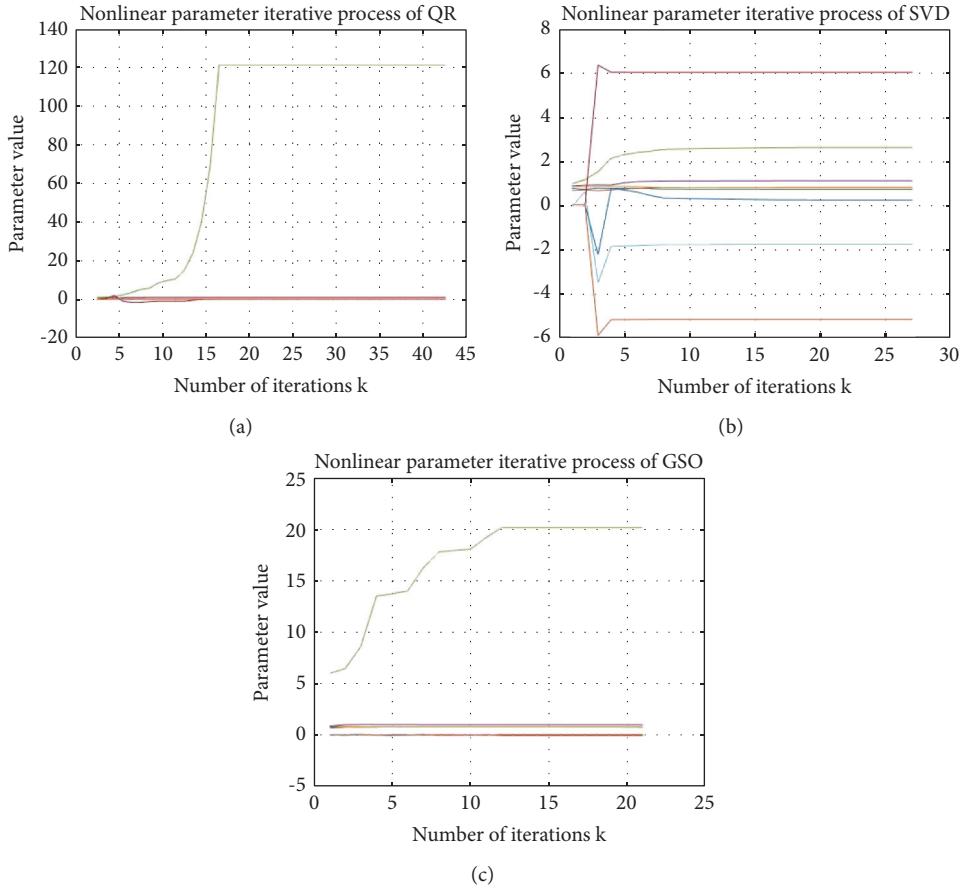

Figure 5: Iterative convergence curves: (a) VP_{QR}, (b) VP_{GSO}, and (c) VP_{SVD}.

4. Conclusion

This paper reports a variable projection algorithm to solve SNLLS problems. In the solution, the Jacobian matrix was decomposed to improve the iteration efficiency and stability of the algorithm. When solving the linear parameters, the Tikhonov regularization method was used to control the ill-conditioning of the model parameters. The RBF neural network data were used for the experiments, and the algorithm was verified through qualitative and quantitative indicators. The experimental results were compared with those obtained by using the variable projection-based matrix nondecomposition method. The experimental results showed that the proposed methods are more efficient than the matrix nondecomposition method. The three proposed methods have their own advantages. The VP_{QR} algorithm is the best in terms of the accuracy, the VP_{GSO} algorithm has the highest computational efficiency, and the VP_{SVD} algorithm is the best in terms of the prediction point residual stability.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant no. 42074009).

References

[1] G. H. GOLUB and V. PEREYRA, "The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate," *SIAM Journal on Numerical Analysis*, vol. 10, no. 2, pp. 413–432, 1973.

[2] Y. Chen, "etc. Accuracy verification and evaluation of Sentinel-1A repeat track differential interferometric synthetic aperture radar in monitoring mining subsidence," *Journal of Applied Remote Sensing*, vol. 14, no. 1, 2020.

[3] V. Haggan-Ozaki, Y. Toyoda, and T. Ozaki, "An Akaike state-space controller for RBF-ARX models," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 1, pp. 191–198, 2009.

[4] L. Kaufman, "A variable projection method for solving separable nonlinear least squares problems," *BIT Numerical. Mathematics*, vol. 15, pp. 49–57, 1975.

[5] A. RUHE and P. A. WEDIN, "Algorithms for separable nonlinear least squares problems," *SIAM Review*, vol. 22, no. 3, pp. 318–337, 1980.
[6] M. Gan, "etc. Gradient radial basis function based baryingcoefficient autoregressive model for nonlinear and nonstationary time series," *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 809–812, 2015.
[7] P. Dianne, O. ' leary, and B. W. Rust, "Variable projection for nonlinear least squares problems," *Computational Optimization and Applications*, vol. 54, no. 3, pp. 579–593, 2013.
[8] A. E. B. Ruano, D. I. Jones, and P. J. Fleming, "A new formulation of the learning problem of a neural network controller," in *Proceedings of the 30th IEEE Conference on Decision and Control*, pp. 865-866, Brighton, U.K., 1991.
[9] M. Gan, "Etc. On some separated algorithms for separable nonlinear least squares problems," *IEEE Transactions on Cybernetics*, vol. 48, no. 10, pp. 2866–2874, 2018.
[10] L. Wang, "Etc. Application of variable projection method based on gram-schmidt orthogonalization in spatial cartesian coordinate transformation model," *KSCE Journal of Civil Engineering*, vol. 23, no. 12, pp. 5194–5200, 2019.
[11] M. Zhai, "etc. A novel characteristic value correction iteration method," *Communications in Statistics - Simulation and Computation*, vol. 48, no. 1, pp. 591–600, 2019.
[12] K. Wang, "Etc. Efficient parameters estimation method for the separable nonlinear least squares problem," *Complexity*, vol. 2020, 10 pages, 2020.
[13] H. Li and Z. Liu, "A probabilistic neural-fuzzy learning system for stochastic modeling," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 4, pp. 898–908, 2008.
[14] G. Golub and V. Pereyra, "Separable nonlinear least squares: the variable projection method and its applications," *Inverse Problems*, vol. 19, no. 2, pp. 1–26, 2003.
[15] R. Zhang, "Proportional functional coefficient time series models," *Journal of Statistical Planning and Inference*, vol. 139, no. 3, pp. 749–763, 2009.