Hai-Rong Cui; Jing Lin; Jian-Nan Su

Multi-step-length gradient iterative method for separable nonlinear least squares problems

Kybernetika, Vol. 60 (2024), No. 2, 197–209

Persistent URL: http://dml.cz/dmlcz/152415

KYBERNETIKA — VOLUME *60* (2024), NUMBER 2, PAGES 197–209

MULTI-STEP-LENGTH GRADIENT ITERATIVE METHOD FOR SEPARABLE NONLINEAR LEAST SQUARES PROBLEMS

Hai-Rong Cui, Jing Lin and Jian-Nan Su

Separable nonlinear least squares (SNLLS) problems are critical in various research and application fields, such as image restoration, machine learning, and system identification. Solving such problems presents a challenge due to their nonlinearity. The traditional gradient iterative algorithm often zigzags towards the optimal solution and is sensitive to the initial guesses of unknown parameters. In this paper, we improve the convergence rate of the traditional gradient method by implementing a multi-step-length gradient iterative algorithm. Moreover, we incorporate the variable projection (VP) strategy, taking advantage of the separable structure observed in SNLLS problems. We propose a multi-step-length gradient iterative-based VP (Mul-GI-VP) method to solve such nonlinear optimization problems. Our simulation results verify the feasibility and high efficiency of the proposed algorithm.

Keywords: separable nonlinear least squares, multi-step-length gradient iterative method, variable projection algorithm, image restoration

Classification: 49M99

1. INTRODUCTION

Nonlinear regression often arises in many research fields, taking the form of a linear combination of nonlinear functions:

$$y(\boldsymbol{a}, \boldsymbol{c}; \boldsymbol{x}) = \sum_{j=1}^{n} c_j f_j(\boldsymbol{a}; \boldsymbol{x}), \tag{1}$$

where $\{f_j, j = 1, \ldots, n\}$ are the nonlinear functions with respect to an input vector $\boldsymbol{x} \in R^{d_{in}}$ whose dimension d_{in} depends on the dataset used. Note that unless otherwise specified, uppercase and lowercase bold symbols in the following denote matrices and vectors respectively. The parameters in (1) can be partitioned into two parts, in which $\boldsymbol{c} = (c_1, \ldots, c_n)^{\mathrm{T}}$ appears linearly and $\boldsymbol{a} \in R^q$ represents the nonlinear parameters. Assuming that the noise is Gaussian, fitting model (1) to m input-output pairs $\{(\boldsymbol{x}_i, y_i), i = 1, \ldots, m\}$ in a least-squares sense is a so-called separable nonlinear least

DOI: 10.14736/kyb-2024-2-0197

squares (SNLLS) problem:

$$\min_{\boldsymbol{a},\boldsymbol{c}} \frac{1}{2} \sum_{i=1}^{m} (y_i - \sum_{j=1}^{n} c_j f_j(\boldsymbol{a}; \boldsymbol{x}_i))^2. \tag{2}$$

Define $\boldsymbol{y} =: (y_1, \ldots, y_m)^{\mathrm{T}}$, $\Phi(\boldsymbol{a})_{ij} =: f_j(\boldsymbol{a}; \boldsymbol{x}_i)$, the optimization problem (2) can be rewritten in a matrix form:

$$\min_{\boldsymbol{a},\boldsymbol{c}} r(\boldsymbol{a}, \boldsymbol{c}) = \frac{1}{2} \left\| \boldsymbol{y} - \Phi(\boldsymbol{a})\boldsymbol{c} \right\|^2, \tag{3}$$

where $\|\cdot\|$ denotes 2-norm of a vector.

The SNLLS problem is commonly encountered in various application fields. For example, in image processing, the blind deconvolution problem [7, 8] requires solving problem (3) to estimate the blind operator. In machine learning, certain types of neural networks (e. g., multi-layer perceptrons and radial-basis function (RBF) networks) are linear combinations of nonlinear active functions [15, 16, 30]. The Prony's method [27], which is the sum of complex exponentials, has been widely used to analyze the frequency components of a signal in signal processing. In system identification, the parameter identification of separable nonlinear models (e. g., Exp-AR model [19], RBF-AR model [28], and BFM-FCAR model [3]) can be formulated as solving SNLLS problems.

The optimization problem (3) is essentially non-convex, making it a challenge to solve such problems. The gradient iterative (GI) algorithm [11, 31, 35], which requires less computational effort, is commonly used to solve such problems. However, the zigzag convergence phenomenon towards the optimal value can often result in a low convergence rate of the GI algorithm. Many improvements have been reported in the literature to enhance the convergence rate of the GI algorithm [5, 10, 21, 23]. These improvements can be roughly classified into two categories. The first is to adjust the search direction, such as the conjugate gradient method developed by Hussu [21] for optimal control problems. The other approach is to select an appropriate step-length [5, 23]. Most of these modifications assume that the step-length is constant, regardless of the dimension of the unknown parameters. To address the problem where some initial variables are far from the true values, while others are not, Chen *et al.* [6] proposed a multi-step-length gradient iterative algorithm for the equation-type model. By introducing multiple step lengths, the iterative algorithm significantly boosts the convergence rate. We utilize the high efficiency of this method and reformulate the optimization problem while introducing a multi-step-length gradient algorithm to update the parameters.

Moreover, the separable structure present in the nonlinear optimization problem (3) allows for an efficient optimization strategy known as variable projection (VP). Variable projection was first proposed by Golub and Pereyra [17, 18] and has been widely used in various application fields [2, 12, 14, 24, 32]. The fundamental idea behind VP optimization strategy is to reduce the dimensionality of parameters by eliminating the linear parameters. Compared with other optimization strategies (e. g., joint optimization method [4, 26], alternating least squares (ALS) method [26], and others), the VP method is more valuable in solving SNLLS problems. This idea of separability has also been widely extended to identify various models in system modeling [9, 25].

In this paper, we analyze the local structure of the optimization problem (3) and propose a multi-step-length gradient iterative-based VP (Mul-GI-VP) method to optimize the parameters of the nonlinear regression problem. The proposed algorithm initially utilizes the VP strategy to remove the linear parameters appearing in (3) and then employs the multi-step-length gradient method to optimize the reduced function containing only nonlinear parameters. As the proposed algorithm only utilizes first-order information, its implementation is more convenient. Furthermore, the incorporation of multi-step-length allows the proposed algorithm to achieve a faster convergence rate.

The remainder of this paper is organized as follows. Section 2 introduces the basic concepts, including the gradient method, multi-step-length, and the VP strategy. In Section 3, we propose the Mul-GI-VP algorithm for solving the SNLLS problem. Section 4 presents numerical experiments on image restoration and parameter estimation of a complex exponential model. Finally, we summarize the main conclusions of this paper in Section 5.

2. PRELIMINARIES

2.1. Multi-step-length gradient algorithm

Consider the quadratic problem

$$v(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^{\mathrm{T}}\mathbf{Q}\boldsymbol{\theta} - \boldsymbol{\eta}^{\mathrm{T}}\boldsymbol{\theta}, \tag{4}$$

where $\mathbf{Q}$ is a symmetric and positive define matrix, and thus can be decomposed as $\mathbf{Q} = \mathbf{J}^{\mathrm{T}}\mathbf{J}$. Many optimization problems in system identification can be formulated as the form of (4). For example, the identification of equation-error type (EET) model [6]

$$\mathbf{A}(z)y(t) = \mathbf{B}(z)u(t) + \boldsymbol{\varepsilon}(t) \tag{5}$$

where $y(t)$ and $u(t)$ are the output and input of the system, $\mathbf{A}(z)$ and $\mathbf{B}(z)$ are the polynomials in the backward-shift operator, denoted as z. Given observations $\{(u_1, y_1), \ldots, (u_L, y_L)\}$, we can obtain the EET model by solving the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\theta}} v(\boldsymbol{\theta}) &= \frac{1}{2}\left\| \boldsymbol{Y}(L) + \sum_{i=1}^{p} \boldsymbol{h}_i(L)\boldsymbol{\theta}_i + \sum_{j=p+1}^{q} \boldsymbol{h}_j(L)\boldsymbol{\theta}_j \right\|^2 \\ &= \frac{1}{2}\left\| \boldsymbol{Y}(L) - \mathbf{M}(L)\boldsymbol{\theta} \right\|^2 \end{aligned} \tag{6}$$

where $\boldsymbol{Y}(L) = (y_1, \ldots, y_L)^{\mathrm{T}}$ are the collected output from 1 to L, $\boldsymbol{\theta} \in R^q$, $\mathbf{M}(L) = [\boldsymbol{h}_1(L), \ldots, \boldsymbol{h}_p(L), \boldsymbol{h}_{p+1}(L), \ldots, \boldsymbol{h}_q(L)]$ in which $\boldsymbol{h}_i(L) = [-y(1-i), -y(2-i), \ldots, -y(L-i)]^{\mathrm{T}}$ for $i \in \{1, \ldots, p\}$ and $\boldsymbol{h}_i(L) = [u(1+p-i+1), u(2+p-i+1), \ldots, u(L+p-i+1)]^{\mathrm{T}}$ for $i \in \{p+1, \ldots, q\}$.

For such problems, Chen et al. [6] proposed a multi-step-length gradient iterative algorithm to improve the convergence rate of the standard GI algorithm. They introduced an auxiliary model

$$\boldsymbol{Y}(L) = \tilde{\mathbf{M}}(L)\boldsymbol{\beta} + \boldsymbol{\varepsilon}(L) \tag{7}$$

where $\boldsymbol{\beta} \in R^q$ are the auxiliary variables and $\tilde{\mathbf{M}}(L)$ is obtained by the Gram-Schmidt procedure, that is,

$$[\boldsymbol{h}_1(L), \ldots, \boldsymbol{h}_q(L)] = \left[\tilde{\mathbf{M}}_1(L), \ldots, \tilde{\mathbf{M}}_q(L)\right] \mathbf{W} \tag{8}$$

$$\mathbf{W} = \begin{bmatrix} 1 & w_{1,2} & \cdots & w_{1,q} \\ 0 & 1 & \cdots & \vdots \\ \vdots & \vdots & \ddots & w_{q-1,q} \\ 0 & 0 & \cdots & 1 \end{bmatrix}, \tag{9}$$

$$w_{i,j} = \frac{\boldsymbol{h}_i^{\mathrm{T}}(L)\tilde{\mathbf{M}}_j(L)}{\tilde{\mathbf{M}}_j^{\mathrm{T}}(L)\tilde{\mathbf{M}}_j(L)}, j < i.$$

The Mul-GI algorithm can be summarized as:

$$\boldsymbol{\beta}^{k+1} = \boldsymbol{\beta}^k + \boldsymbol{\lambda}^k \boldsymbol{d}^k, \tag{10}$$

$$\begin{aligned} \boldsymbol{d}^k = -[&\tilde{\mathbf{M}}_1^{\mathrm{T}}(L)\tilde{\mathbf{M}}_1(L)\beta_1^k - \tilde{\mathbf{M}}_1^{\mathrm{T}}(L)\boldsymbol{Y}(L), \ldots, \\ &\tilde{\mathbf{M}}_q^{\mathrm{T}}(L)\tilde{\mathbf{M}}_q(L)\beta_q^k - \tilde{\mathbf{M}}_q^{\mathrm{T}}(L)\boldsymbol{Y}(L)]^{\mathrm{T}}, \end{aligned} \tag{11}$$

$$\boldsymbol{\lambda}^k = \mathrm{diag}(\lambda_1^k, \lambda_2^k, \ldots, \lambda_q^k), \tag{12}$$

where $\lambda_i^k = [\tilde{\mathbf{M}}_i^{\mathrm{T}}(L)\tilde{\mathbf{M}}_i(L)]^{-1}$ is the step-length of the direction d_i^k (i. e., the i_{th} element in $\boldsymbol{d}^k$). The parameter $\boldsymbol{\theta}$ can be calculated after the auxiliary variables $\boldsymbol{\beta}$ are updated

$$\boldsymbol{\theta}^{k+1} = \mathbf{W}^{-1}\boldsymbol{\beta}^{k+1}. \tag{13}$$

Compared with the standard gradient method, the multi-step-length gradient iterative algorithm can achieve faster convergence rate and is less sensitive to the initial values of the estimated parameters [6].

2.2. Variable projection method

The separable structure presented in problem (3) allows for an efficient optimization strategy, the VP strategy, which was proposed by Golub Pereyra [18]. The underlying principle of the VP strategy is to reduce the dimensionality of the parameters by solving a linear least-squares subproblem of (3)

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{\Phi}(\boldsymbol{a})\boldsymbol{c}\|^2 = \boldsymbol{\Phi}(\boldsymbol{a})^{\dagger}\boldsymbol{y}, \tag{14}$$

where $\boldsymbol{\Phi}^{\dagger} = (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}$ is the Moore-Penrose inverse of matrix $\boldsymbol{\Phi}$. Plugging (14) into (3) yields a reduced function

$$r_1(\boldsymbol{a}) = \frac{1}{2} \left\|\boldsymbol{y} - \boldsymbol{\Phi}(\boldsymbol{a})\boldsymbol{\Phi}(\boldsymbol{a})^{\dagger}\boldsymbol{y}\right\|^2 = \frac{1}{2} \left\|\mathbf{P}_{\boldsymbol{\Phi}(\boldsymbol{a})}^{\perp}\boldsymbol{y}\right\|^2, \tag{15}$$

where $\mathbf{P}_{\mathbf{\Phi}}^{\perp} = \mathbf{I} - \mathbf{\Phi}\mathbf{\Phi}^{\dagger}$ is a projection operator that projects a vector onto the orthogonal complement of space spanned by the columns of $\mathbf{\Phi}$.

The VP strategy usually yields a lower dimension of estimated parameter space, a better-conditioned problem [30] and a faster convergence rate of the optimization algorithm [1, 4, 29]. Note that $\hat{\boldsymbol{c}}$ is the optimal solution of the convex problem (3) for fixed $\boldsymbol{a}$, we can obtain the gradient of the reduced function

$$\frac{\partial r_1(\boldsymbol{a})}{\partial \boldsymbol{a}} = \frac{\partial r}{\partial \boldsymbol{a}} + \frac{\partial \boldsymbol{c}}{\partial \boldsymbol{a}^{\mathrm{T}}}\frac{\partial r}{\partial \boldsymbol{c}} = \frac{\partial r(\boldsymbol{a}, \hat{\boldsymbol{c}})}{\partial \boldsymbol{a}} = -(\mathrm{D}\Phi(\boldsymbol{a})\hat{\boldsymbol{c}})^{\mathrm{T}}\mathbf{P}_{\mathbf{\Phi}(\boldsymbol{a})}^{\perp}\boldsymbol{y}, \tag{16}$$

where $\mathrm{D}\mathbf{\Phi}(\boldsymbol{a})$ is the Fréchet derivative of $\mathbf{\Phi}(\boldsymbol{a})$.

3. MULTI-STEP-LENGTH GRADIENT ITERATIVE-BASED VP METHOD

In this section, we analyze the local structure of the reduced problem (15), and propose a multi-step-length gradient based-variable projection (Mul-GI-VP) algorithm by incorporating the advantage of muti-step-length method.

In the following part, the reduced nonlinear regression problem (15) is first approximated by its second-order Taylor expansion

$$r_1(\boldsymbol{a}^{k+1}) \approx r_1(\boldsymbol{a}^k) + r_1'(\boldsymbol{a}^k)^{\mathrm{T}}(\Delta\boldsymbol{a}^k) + \frac{1}{2}(\Delta\boldsymbol{a}^k)^{\mathrm{T}}\mathbf{H}(\boldsymbol{a}^k)\Delta\boldsymbol{a}^k, \tag{17}$$

where $\Delta\boldsymbol{a}^k = \boldsymbol{a}^{k+1} - \boldsymbol{a}^k$ and $\mathbf{H}$ is the Hessian matrix. In practice, the Hessian matrix can be often approximated by different methods. In some settings, calculating Jacobian matrix of reduced function does not require huge calculation. In this case, we can approximate the Hessian matrix by

$$\mathbf{H} \approx \mathbf{Q} = \mathbf{J}^{\mathrm{T}}\mathbf{J}, \tag{18}$$

where $\mathbf{J}$ is the Jacobian matrix of the reduced function. For the reduced problem (15), Golub and Pereyra [18] presented the analytical expression of the Jacobian matrix

$$\mathbf{J}_{\mathrm{GP}} = -\mathbf{P}_{\mathbf{\Phi}}^{\perp}\mathrm{D}\mathbf{\Phi}\mathbf{\Phi}^{-}\boldsymbol{y} - (\mathbf{P}_{\mathbf{\Phi}}^{\perp}\mathrm{D}\mathbf{\Phi}\mathbf{\Phi}^{-})^{\mathrm{T}}\boldsymbol{y}, \tag{19}$$

where $\mathbf{\Phi}^{-}$ is generalized inverse of $\mathbf{\Phi}$ (satisfying $(\mathbf{\Phi}\mathbf{\Phi}^{-})^{\mathrm{T}} = \mathbf{\Phi}\mathbf{\Phi}^{-}$ and $\mathbf{\Phi}\mathbf{\Phi}^{-}\mathbf{\Phi} = \mathbf{\Phi}$) and $\mathrm{D}\mathbf{\Phi}$ is the Fréchet derivative of $\mathbf{\Phi}$. Kaufman [22] proposed a simplified form by dropping the second term of (19)

$$\mathbf{J}_{\mathrm{Kau}} = -\mathbf{P}_{\mathbf{\Phi}}^{\perp}\mathrm{D}\mathbf{\Phi}\mathbf{\Phi}^{-}\boldsymbol{y}. \tag{20}$$

Many empirical evidences [13, 17] suggest that the simplified Jacobian matrix form (20) achieves similar performance to the full form (19). In the following part, we use Kaufman's simplified form of Jacobian matrix instead of Golub and Pereyra's form for convenience, and the second-order approximated function of the reduced function can be expressed as

$$\begin{aligned} v(\boldsymbol{a}^{k+1}) &= r_1(\boldsymbol{a}^k) + r_1'(\boldsymbol{a}^k)^{\mathrm{T}}\Delta\boldsymbol{a}^{k+1} + \frac{1}{2}(\Delta\boldsymbol{a}^{k+1})^{\mathrm{T}}\mathbf{J}_{\mathrm{Kau}}^{\mathrm{T}}(\boldsymbol{a}^{\boldsymbol{k}})\mathbf{J}_{\mathrm{Kau}}(\boldsymbol{a}^{\boldsymbol{k}})\Delta\boldsymbol{a}^{k+1} \\ &= \frac{1}{2}(\boldsymbol{a}^{k+1})^{\mathrm{T}}\mathbf{Q}\boldsymbol{a}^{k+1} - \boldsymbol{\eta}^{\mathrm{T}}\boldsymbol{a}^{k+1} + \gamma \end{aligned} \tag{21}$$

where $\Delta \boldsymbol{a}^{k+1} = \boldsymbol{a}^{k+1} - \boldsymbol{a}^k$, $\mathbf{Q} = \mathbf{J}_{\text{Kau}}^{\text{T}}(\boldsymbol{a}^k)\mathbf{J}_{\text{Kau}}(\boldsymbol{a}^k) = \boldsymbol{y}^{\text{T}}(\boldsymbol{\Phi}^-)^{\text{T}}(\text{D}\boldsymbol{\Phi})^{\text{T}}\mathbf{P}_{\boldsymbol{\Phi}}^{\perp}\text{D}\boldsymbol{\Phi}\boldsymbol{\Phi}^-\boldsymbol{y}$, $\boldsymbol{\eta} = -r_1'(\boldsymbol{a}^k) + \mathbf{Q}\boldsymbol{a}^k$ and $\gamma = r_1(\boldsymbol{a}^k) - r_1'(\boldsymbol{a}^k)^{\text{T}}\boldsymbol{a}^k + \frac{1}{2}(\boldsymbol{a}^k)^{\text{T}}\mathbf{Q}\boldsymbol{a}^k$. Note that the $r_1'(\boldsymbol{a}^k)$ computed by (16) is equal to $\mathbf{J}_{\text{Kau}}^{\text{T}}(\boldsymbol{a}^k)\mathbf{P}_{\boldsymbol{\Phi}(\boldsymbol{a}^k)}^{\perp}\boldsymbol{y}$, since $(\mathbf{P}_{\boldsymbol{\Phi}}^{\perp})^{\text{T}} = \mathbf{P}_{\boldsymbol{\Phi}}^{\perp}$ and $(\mathbf{P}_{\boldsymbol{\Phi}}^{\perp})^2 = \mathbf{P}_{\boldsymbol{\Phi}}^{\perp}$.

For large-scale problems, the approximated Hessian matrix can be obtained by a recursive Broyden, Fletcher, Goldfarb, and Shanno (BFGS) method that just uses the gradient information of the reduced problem

$$\mathbf{H}^k = \left(\mathbf{I} - \frac{\Delta \boldsymbol{a}^k(\Delta \boldsymbol{g}^k)^{\text{T}}}{(\Delta \boldsymbol{g}^k)^{\text{T}}\Delta \boldsymbol{a}^k}\right)\mathbf{H}^{k-1} \cdot \left(\mathbf{I} - \frac{\Delta \boldsymbol{g}^k(\Delta \boldsymbol{a}^k)^{\text{T}}}{(\Delta \boldsymbol{g}^k)^{\text{T}}\Delta \boldsymbol{a}^k}\right) + \frac{\Delta \boldsymbol{a}^k(\Delta \boldsymbol{a}^k)^{\text{T}}}{(\Delta \boldsymbol{g}^k)^{\text{T}}\Delta \boldsymbol{a}^k}, \tag{22}$$

$$\Delta \boldsymbol{a}^k = \boldsymbol{a}^k - \boldsymbol{a}^{k-1}, \tag{23}$$

$$\Delta \boldsymbol{g}^k = \boldsymbol{g}^k - \boldsymbol{g}^{k-1}, \tag{24}$$

where $\boldsymbol{a}^k$ is the estimated value of nonlinear parameter and $\boldsymbol{g}^k$ is the gradient of the reduced function at kth iteration. In this case, $\mathbf{Q}$, $\boldsymbol{\eta}$ and γ in the second-order approximation function are

$$\begin{aligned}
&\mathbf{Q} = \mathbf{H}^k = \mathbf{J}_{\text{BFGS}}^{\text{T}}(\boldsymbol{a}^k)\mathbf{J}_{\text{BFGS}}(\boldsymbol{a}^k),\\
&\boldsymbol{\eta} = -r_1'(\boldsymbol{a}^k) + \mathbf{H}^k\boldsymbol{a}^k,\\
&\gamma = r_1(\boldsymbol{a}^k) - r_1'(\boldsymbol{a}^k)^{\text{T}}\boldsymbol{a}^k + \frac{1}{2}(\boldsymbol{a}^k)^{\text{T}}\mathbf{H}^k\boldsymbol{a}^k,
\end{aligned}$$

where $\mathbf{J}_{\text{BFGS}}$ is obtained by performing Cholesky decomposition on $\mathbf{H}^k$.

The minimization of approximated function is obviously a quadratic optimization problem, then the multi-step-length gradient iterative method is employed to update the nonlinear parameters. Comparing (21) with (6), we can find that

$$\mathbf{M} = \mathbf{J}(\boldsymbol{a}^k), \mathbf{Y} = \mathbf{J}(\boldsymbol{a}^k)\boldsymbol{a}^k - \mathbf{P}_{\boldsymbol{\Phi}(\boldsymbol{a}^k)}^{\perp}\boldsymbol{y}, \tag{25}$$

where $\mathbf{J}$ can be $\mathbf{J}_{\text{Kau}}$ or $\mathbf{J}_{\text{BFGS}}$. Applying the Gram-Schmidt procedure to $\mathbf{M}$, we can obtain $\tilde{\mathbf{M}}$ and $\mathbf{W}$. Then executing the Mul-GI algorithm (10)-(13), the nonlinear parameters can be updated by:

$$\boldsymbol{a}^{k+1} = \mathbf{W}^{-1}\boldsymbol{\beta}^{k+1}. \tag{26}$$

This minimization strategy for solving the nonlinear regression problem is named as *multi-step-length gradient iterative-based variable projection* (Mul-GI-VP) method. The proposed algorithm first utilizes the VP method to eliminate the linear parameters and then the multi-step-length gradient iterative method is adopted to update the nonlinear parameters. The Mul-GI-VP algorithm is summarized in Algorithm 1:

Algorithm 1: Multi-step-length gradient iterative-based variable projection (Mul-GI-VP) algorithm

Initialize

Choose positive numbers ε_1 and ε_2, a maximum iteration number N, and the initial values of nonlinear parameters $\boldsymbol{a}^0$. Let $k = 0$ and the initial $\boldsymbol{\beta}^0 = \mathbf{1}/10^6$, with $\mathbf{1}$ being a q-dimensional column vector.

Estimate nonlinear parameters a
Step 1: Eliminate the linear parameters $\hat{\boldsymbol{c}}$ by (14) for the fixed $\boldsymbol{a} = \boldsymbol{a}^k$, resulting in a reduced function (15) that only contains the nonlinear parameters.
Step 2: Compute the approximated Hessian matrix according to Kaufman's simplified Jacobian matrix (20) or using the BFGS method (22)-(24).
Step 3: Utilize the quadratic function (21) to approximate the reduced function by computing the symmetric positive matrix $\mathbf{Q} = \mathbf{J}_{\text{Kau}}^{\text{T}}\mathbf{J}_{\text{Kau}}$ or $\mathbf{J}_{\text{BFGS}}^{\text{T}}\mathbf{J}_{\text{BFGS}}$, $\boldsymbol{\eta}$ and γ.
Step 4: Compute $\mathbf{M}$ and $\mathbf{Y}$ by (25) using $\mathbf{J}_{\text{Kau}}$ or $\mathbf{J}_{\text{BFGS}}$ and then apply the Gram-Schmidt procedure to $\mathbf{M}$ to obtain $\widetilde{\mathbf{M}}$ and $\mathbf{W}$ by (8) and (9).
Step 5: Compute the update direction $\boldsymbol{d}^k$ by (11) and $\boldsymbol{\lambda}^k$ by (12).
Step 6: Calculate the auxiliary variables $\boldsymbol{\beta}^k$ by (10) and then update the nonlinear parameters $\boldsymbol{a}^{k+1}$ by (26).
Step 7: if $\left|r_1(\boldsymbol{a}^{k+1}) - r_1(\boldsymbol{a}^k)\right| < \varepsilon_1$ or $\left\|\boldsymbol{a}^{k+1} - \boldsymbol{a}^k\right\| < \varepsilon_2$
  Terminate the process, break;
else
  $k = k + 1$, go to **Step 1**;
end
Output
Nonlinear parameters: $\hat{\boldsymbol{a}} = \boldsymbol{a}^{k+1}$;
Linear parameters: $\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{c}} \frac{1}{2}\left\|\boldsymbol{y} - \boldsymbol{\Phi}(\hat{\boldsymbol{a}})\boldsymbol{c}\right\|^2 = \boldsymbol{\Phi}(\hat{\boldsymbol{a}})^{\dagger}\boldsymbol{y}$.

4. NUMERICAL EXPERIMENTS

In this section, two numerical experiments, including image restoration and data fitting of complex exponential model, are employed to verify the performance of the proposed algorithm. Different optimization strategies (including the joint optimization method, alternating optimization method and VP method) are used as comparison methods to verify the effectiveness of the proposed algorithm. To ensure the fairness of comparison, all the algorithms compared in the experiments adopt the same randomly generated initial values and all the algorithms were run on MATLAB R2018b.

4.1. Data fitting of complex exponential model

In this subsection, we consider the following complex exponential model

$$f(\boldsymbol{a}, \boldsymbol{c}; t) = c_1 e^{-a_2 t}\cos(a_3 t) + c_2 e^{-a_1 t}\cos(a_2 t) + \epsilon_t, \tag{27}$$

where $\boldsymbol{a} = (a_1, a_2, a_3)^{\text{T}}$ and $\boldsymbol{c} = (c_1, c_2)^{\text{T}}$ are the nonlinear parameters and linear parameters; ϵ_t is the white noise.

Here the parameters in (27) are set to be: $\boldsymbol{a} = (1, 2.5, 4)^{\text{T}}$ and $\boldsymbol{c} = (6, 1)^{\text{T}}$. The noise $\{\epsilon_t\}$ obeys Gaussian distribution with 0 mean and variance $\sigma^2 = 0.1^2$. Based on these selected parameters, 19 data points from $t = 0.1$ to 1 with time interval 0.05 are randomly generated.

Assume that the form of model (27) is known, the parameter estimation is equivalent

to solving the following nonlinear regression problem according the observations $\{(t_i, y_i)\}$

$$\frac{1}{2}\sum_{i=1}^{m}(y_i - c_1 e^{-a_2 t_i}\cos(a_3 t_i) - c_2 e^{-a_1 t_i}\cos(a_2 t_i))^2. \tag{28}$$

Three different optimization strategies are used to solve the problem (28), and for each strategy, we use the standard gradient iterative method and multi-step-length gradient iterative method. For a fair comparison, the initial values of each algorithm are kept the same. Figure 1 shows the comparison of the convergence process of different algorithms. The error of estimated parameters is defined as:

$$\mathrm{Err}(\boldsymbol{a}^k) = \left\|\boldsymbol{a}^k - \boldsymbol{a}_{\mathrm{true}}\right\|^2,$$

where $\boldsymbol{a}^k$ are the estimated nonlinear parameters at kth iteration. From Figure 1, we can observe that the multi-step-length gradient iterative algorithm outperforms the standard gradient iterative method, achieving faster convergence rate. The VP optimization strategy greatly accelerates the convergence speed of the algorithm and makes the parameter estimation more accurate. The overall comparison results confirm the effectiveness of the proposed Mul-GI-VP algorithm.

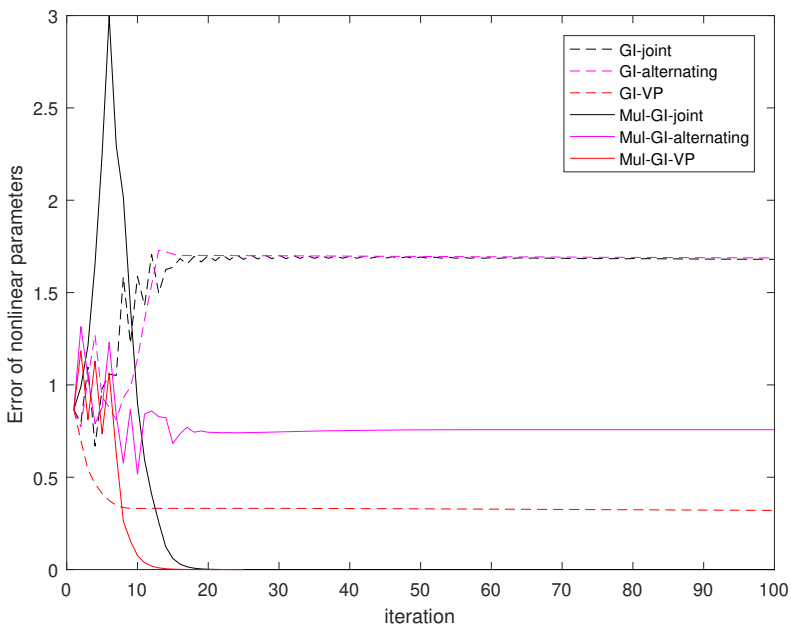


Fig. 1. Convergence comparison of different algorithms for estimating the parameters of complex exponential model.

4.2. Image restoration

In image restoration, a type of image deblurring problem, named *blind deconvolution*, can be reduced to solving an SNLLS problem. The image blurring process can be modeled by

$$\boldsymbol{b} = \mathbf{A}(\mathbf{P}(\boldsymbol{a}))\boldsymbol{x}_{\text{true}} + \boldsymbol{\epsilon}, \tag{29}$$

where $\boldsymbol{b}$ represents the blurred image, $\mathbf{A}(\mathbf{P}(\boldsymbol{a}))$ is a blurring operator that is determined by the point spread function (PSF) $\mathbf{P}(\boldsymbol{a})$ and the boundary condition. Various PSFs have been used in real applications. In this case, we use the Gauss blur $\mathbf{P}(\boldsymbol{a})$, which can be expressed as

$$p_{ij} = \exp\left(-\frac{1}{2}\begin{bmatrix} i-k \\ j-l \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} s_1^2 & r^2 \\ r^2 & s_2^2 \end{bmatrix}^{-1} \begin{bmatrix} i-k \\ j-l \end{bmatrix}\right),$$

where (k, l) is the center of PSF. The PSF is determined by the orientation and spread parameters $\boldsymbol{a} = (s_1, s_2, r)^{\mathrm{T}}$, which is regarded as the nonlinear parameters of the blurring model. We aim to estimate the parameters $\boldsymbol{a}$ to obtain the blurring operator , and then to reconstruct the true image $\boldsymbol{x}$, which is equivalent to solving the nonlinear regression problem

$$\min_{\boldsymbol{a},\boldsymbol{x}} \frac{1}{2} \left\|\boldsymbol{b} - \mathbf{A}(\mathbf{P}(\boldsymbol{a}))\boldsymbol{x}\right\|^2, \tag{30}$$

where we assume that the image has a reflexive boundary. More information about the blurring model can refer to [7, 20].

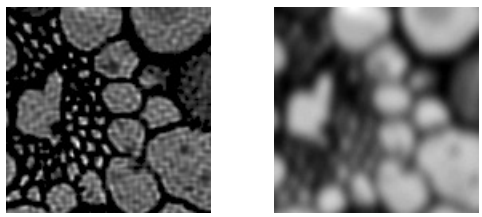


Fig. 2. A piece of grain with 128×128 size and the blurred image generated according to blurring model (1% Gaussian noise is added) (29).

In this case, a piece of grain with size of 128×128 shown in Figure 2 is used and the blurred image generated by model (29) (1% Gaussian noise is added) is shown in Figure 2. Different methods (including GI-joint, Mul-GI-joint, GI-alternating, Mul-GI-alternating, GI-VP and Mul-GI-VP) are employed to reconstruct the true image. Figure 3 shows the convergence process of the objective function values obtained by different methods. As shown in Figure 3, we can find that: 1) multi-step-length optimization strategy makes the algorithm converge faster; 2) the proposed Mul-GI-VP method obviously

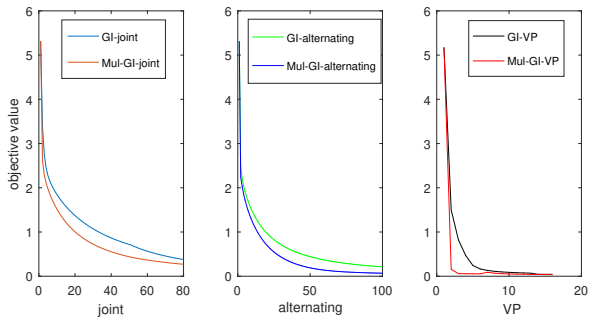
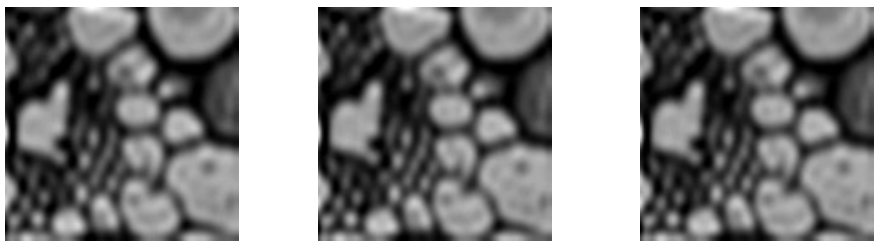
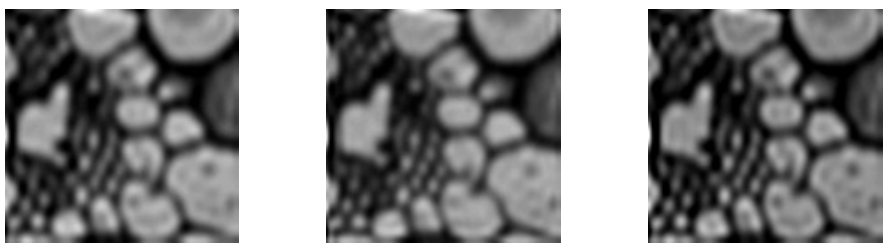


Fig. 3. Comparison of convergence of different methods.



(a) GI-joint (b) Mul-GI-joint (c) GI-alternating



(d) Mul-GI-alternating (e) GI-VP (f) Mul-GI-VP

Fig. 4. Recovered images by different methods.

outperforms the other methods, especially the joint optimization method and alternating optimization method.

The above results confirm the effectiveness of the proposed Mul-GI-VP algorithm compared to the other first-order gradient methods.

5. CONCLUSION

In this paper, we discuss gradient iterative-based algorithms for SNLLS problems. Taking advantage of the special structure presented in such models, we propose a highly efficient Mul-GI-VP algorithm. This method utilizes the VP strategy to obtain a reduced problem with a lower dimension of estimated parameters and then introduces the multi-step-length strategy to update the nonlinear parameters. In this way, the proposed algorithm overcomes the low convergence rate of the standard gradient iterative algorithm and is more robust to the initial values of parameters.

Multi-step-length provides a feasible way to improve the convergence rate of the gradient iterative method. For some real-world problems [33, 34], when the parameter dimension is large, it will be a meaningful research topic in the future to explore how to use as few step-lengths as possible to achieve the best performance.

(Received July 26, 2023)

REFERENCES

[1] A. Y. Aravkin, D. Drusvyatskiy, and T. van Leeuwen: Efficient quadratic penalization through the partial minimization technique. IEEE Trans. Automat. Control *63* (2017), 7, 2131–2138. DOI:10.1109/TAC.2017.2754474

[2] G. Y. Chen, M. Gan, C. P. Chen, and L. Chen: A two-stage estimation algorithm based on variable projection method for gps positioning. IEEE Trans. Instrument. Measur. *67* (2018), 11, 2518–2525.

[3] G. Y. Chen, M. Gan, C. P. Chen, and H. X. Li: Basis function matrix-based flexible coefficient autoregressive models: A framework for time series and nonlinear system modeling. IEEE Trans. Cybernet. *51* (2021), 2, 614–623. DOI:10.1109/TCYB.2019.2900469

[4] G. Y. Chen, S. Q. Wang, M. Gan, and C. P. Chen: Insights into algorithms for separable nonlinear least squares problems. IEEE Trans. Imagw Process. *30* (2021), 2, 1207–1218. DOI:10.1109/TIP.2020.3043087

[5] J. Chen and F. Ding: Modified stochastic gradient identification algorithms with fast convergence rates. J. Vibration Control *17* (2011), 9, 1281–1286. DOI:10.1177/1077546310376989

[6] J. Chen, F. Ding, Y. Liu, and Q. Zhu: Multi-step-length gradient iterative algorithm for equation-error type models. Systems Control Lett. *115* (2018), 15–21. DOI:10.1016/j.sysconle.2018.03.003

[7] J. Chung and J. G, Nagy: An efficient iterative approach for large-scale separable nonlinear inverse problems. SIAM J. Scientif. Comput. *31* (2010), 6, 4654–4674. DOI:10.1137/080732213

[8] A. Cornelio, E. L. Piccolomini, and J. G. Nagy: Constrained numerical optimization methods for blind deconvolution. Numer. Algorithms *65* (2014) 1, 23–42. DOI:10.1007/s11075-013-9693-z

[9] F. Ding: Coupled-least-squares identification for multivariable systems. IET Control Theory Appl. *7* (2013), 1, 68–79. DOI:10.1049/iet-cta.2012.0171

[10] F. Ding, G. Liu, and X. P. Liu: Partially coupled stochastic gradient identification methods for non-uniformly sampled systems. IEEE Trans. Automat. Control *55* (2010), 8, 1976–1981. DOI:10.1109/TAC.2010.2050713

[11] F. Ding, Y. Liu, and B. Bao: Gradient-based and least-squares-based iterative estimation algorithms for multi-input multi-output systems. Proc. Inst. Mechanic. Engrs, Part I: J. Systems Control Engrg. *226* (2012), 1, 43–55. DOI:10.1177/0959651811409491

[12] N. B. Erichson, P. Zheng, K. Manohar, S. L. Brunton, J. N. Kutz, and A. Y. Aravkin: Sparse principal component analysis via variable projection. SIAM J. Appl. Math. *80* (2020), 2, 977–1002. DOI:10.1137/18M1211350

[13] M. Gan, C. P. Chen, G. Y. Chen, and L. Chen: On some separated algorithms for separable nonlinear least squares problems. IEEE Trans. Cybernet. *48* (2018), 10, 2866–2874. DOI:10.1109/TCYB.2017.2751558

[14] M. Gan, Y. Guan, G. Y. Chen, and C. P. Chen: Recursive variable projection algorithm for a class of separable nonlinear models. IEEE Trans. Neural Netw. Learn. Syst. (2020).

[15] M. Gan, H. X. Li, and H. Peng: A variable projection approach for efficient estimation of rbf-arx model. IEEE Trans. Cybernet. *45* (2014), 3, 462–471. DOI:10.1109/TSMC.2014.2358635

[16] M. Gan and H. Peng: Stability analysis of rbf network-based state-dependent autoregressive model for nonlinear time series. Appl. Soft Comput. *12* (2012), 1, 174–181. DOI:10.1016/j.asoc.2011.08.055

[17] G. Golub and V. Pereyra: Separable nonlinear least squares: the variable projection method and its applications. Inverse Problems *19* (2003), R1. DOI:10.1088/0266-5611/19/2/201

[18] G. .H Golub and V. Pereyr: The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. SIAM J. Numer. Snal. *10* (1973), 2, 413–432. DOI:10.1137/0710036

[19] V. Haggan and T. Ozaki: Modelling nonlinear random vibrations using an amplitude-dependent autoregressive time series model. Biometrika *68* (1981), 1, 189–196. DOI:10.1093/biomet/68.1.189

[20] P. Ch. Hansen, J. G. Nagy, and D. P. O'leary: Deblurring images: matrices, spectra, and filtering. SIAM 2006.

[21] A Hussu: The conjugate-gradient method for optimal control problems with undetermined final time. Int. J. Control *15* (1972), 1, 79–82. DOI:10.1080/00207177208932130

[22] L. Kaufman: A variable projection method for solving separable nonlinear least squares problems. BIT Numer. Math. *15* (1975), 1, 49–57. DOI:10.1007/BF01932995

[23] J. Li, Y. Zheng, and Z. Lin: Recursive identification of time-varying systems: Self-tuning and matrix rls algorithms. Syst. Control Lett. *66* (2014), 104–110. DOI:10.1016/j.sysconle.2014.01.004

[24] M. Li, A. Abubakar, F. Gao, and T. M Habashy: Application of the variable projection scheme for calibration in electromagnetic data inversion. IEEE Trans. Antennas Propag. *64* (2015), 1, :332–335. DOI:10.1109/TAP.2015.2498946

[25] Y. Liu, F. Ding, and Y. Shi: An efficient hierarchical identification method for general dual-rate sampled-data systems. Automatica *50* (2014), 3, 962–970. DOI:10.1016/j.automatica.2013.12.025

[26] T. Okatani and K. Deguchi: On the wiberg algorithm for matrix factorization in the presence of missing components. Int. J. Comput. Vision *72* (2007), 3, 329–337. DOI:10.1007/s11263-006-9785-5

[27] M. R, Osborne and G. K. Smyth: A modified prony algorithm for exponential function fitting. SIAM J. Scient. Comput. *16* (1995), 1, :119–138. DOI:10.1137/0916008

[28] H. Peng, T. Ozaki, Y. Toyoda, H. Shioya, K. Nakano, V. Haggan-Ozaki, and M. Mori: Rbf-arx model-based nonlinear system modeling and predictive control with application to a nox decomposition process. Control Engrg. Practice *12* (2004), 2, 191–203. DOI:10.1016/S0967-0661(03)00050-9

[29] A. Ruhe and Pe. A. Wedin: Algorithms for separable nonlinear least squares problems. SIAM Rev. *22* (1980), 3, 318–337. DOI:10.1137/1022057

[30] J. Sjoberg and M. Viberg: Separable non-linear least-squares minimization-possible improvements for neural net fitting. In: Neural Networks for Signal Processing VII. Proc. 1997 IEEE Signal Processing Society Workshop, IEEE 1997, pp. 345–354.

[31] G. Stathopoulos, M. Korda, and C. N. Jones: Solving the infinite-horizon constrained lqr problem using accelerated dual proximal methods. IEEE Trans. Automat. Control. *62* (2016), 4, 1752–1767. DOI:10.1109/TAC.2016.2594381

[32] H. Yang, J. Gao, and Z. Wu: Blur identification and image super-resolution reconstruction using an approach similar to variable projection. IEEE Signal Process. Lett. *15* (2008), 289–292. DOI:10.1109/LSP.2007.911743

[33] D. Yu, C. P. Chen, and H. Xu: Fuzzy swarm control based on sliding-mode strategy with self-organized omnidirectional mobile robots system. IEEE Trans. Systems Man Cybernet.: Systems *52* (2021), 4, 2262–2274. DOI:10.1109/TSMC.2020.3048733

[34] D. Yu, H. Xu, C. P. Chen, W. Bai, and Z. Wang: Dynamic coverage control based on k-means. IEEE Trans. Industr. Electron. 2021.

[35] J. Zeng, T. He, and M. Wang: A fast proximal gradient algorithm for decentralized composite optimization over directed networks. Systems Control Lett. *107* (2017), 36–43. DOI:10.1016/j.sysconle.2017.07.005

Hai-Rong Cui, College of Computer Science and Technology, Qingdao University, 308 Ningxia Road, Qingdao, 266071, Shandong. P. R. China.
e-mail: cui-hairong@163.com

Jing Lin, China Construction Industrial and Energy Engineering Group Co., Ltd, 210049. P. R. China.
e-mail: paynelin@csces.com

Jian-Nan Su, Corresponding author. College of Computer and Data Science, Fuzhou University, Qishan Road, Fuzhou, 350108, Fujian. P. R. China.
e-mail: sjn.fzu@gmail.com