Math. Model. Nat. Phenom. 21 (2026) 3
https://doi.org/10.1051/mmnp/2026001

Mathematical Modelling of Natural Phenomena
www.mmnp-journal.org

A NEW CLASS OF HIGH-DIMENSIONAL PARTIALLY LINEAR VARYING COEFFICIENT MODEL AND ITS APPLICATIONS

Jie Zhou, Aifen Feng*, Jiaxin Fu,
Zhengfen Jin and Mengmeng Zhao

Abstract. This paper proposes a new method for solving a class of high-dimensional partially linear varying coefficient model. Firstly, an adaptive elastic net regularization term is added to the partially linear varying coefficient model to obtain a new model. This model improves the accuracy of the model by adjusting the parameter weights. Then, the ADMM method is used to solve the model parameters, and the convergence of the algorithm is analyzed. Subsequently, numerical simulations are conducted on high-dimensional datasets to demonstrate the effectiveness of the proposed method. Finally, the proposed method is applied to practical cases for testing and analysis, further verifying its effectiveness and application value.

Mathematics Subject Classification. 90C06, 90C25, 90C30, 62J05.

Received April 4, 2025. Accepted December 31, 2025.

1. Introduction

Entering the era of big data, science and technology and computers are rapidly developing, and data is growing explosively in fields such as machine learning and artificial intelligence, gene expression analysis, brain neural networks, and medical imaging disease diagnosis and treatment risk management. High dimensional data analysis is rapidly developing and has become a popular research field. Models corresponding to high-dimensional data often have complex diversity and high abstraction. Therefore, it is an important task to find high-dimensional regression model with simple structure and easy interpretability. Firstly, Hastine and Tibishirani [1] proposed the varying coefficient model, which demonstrates wider applicability and stronger interpretability. The varying coefficient model can be expressed as:

$$Y = X^T\beta(T) + \varepsilon, \tag{1.1}$$

where Y is the response variable, X and T are covariates, $\beta(T)$ is the vector of unknown coefficient functions, and ε is the model error.

Based on the varying coefficient model, Zhang *et al.* [2] proposed the partially linear varying coefficient model which can be expressed as:

$$Y = X^T\beta + Z^T\alpha(T) + \varepsilon, \tag{1.2}$$

Keywords and phrases: Partially linear varying coefficient model, adaptive elastic net, ADMM, variable selection.

School of Mathematics and Statistics, Henan University of Science and Technology, Luoyang 471023, China.

* Corresponding author: faf@haust.edu.cn

where Y is the response variable, $X \in \mathrm{R}^p$, $Z \in \mathrm{R}^q$, and $T \in \mathrm{R}^1$ are covariates. $\beta = (\beta_1, \beta_2, \cdots, \beta_p)^T$ is a p-dimensional vector of unknown parameters, and $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_q)^T$ is a vector of unknown coefficient functions. ε is the model error with $E(\varepsilon|X, T, Z) = 0$. When $\alpha(T) = \alpha$, where α is a constant vector, model (1.2) represents a linear model. When $q = 1$ and $Z = 1$, model (1.2) transforms into a partially linear model. When $p = 0$, model (1.2) becomes a varying coefficient model.

The partially linear varying coefficient model has strong adaptability because it allows some variables to appear in linear form while others appear in nonlinear form. The model not only reduces the impact of high-dimensional data on computational performance and avoids the curse of dimensionality, but also improves robustness to outliers and noise. Furthermore, partially linear varying coefficient models can effectively handle multicollinearity problems and naturally select variables. These advantages have enabled partially linear varying coefficient models to play important roles in various fields such as economics, finance, biostatistics, and environmental science. Hong *et al.* [3] applied the partially linear varying coefficient model to the problem of commodity sales management. Qiu and Zhou [4] applied partially linear varying coefficient model to analyze Busselton population health survey. Li and Greene [5] proposed a semi parametric partially linear varying coefficient model to study the treatment of chronic kidney disease. Wu *et al.* [6] used a partially linear varying coefficient model to consider the interaction between gene and environment.

When dealing with varying coefficient models, the main methods include polynomials, splines, Huber functions, and kernel-weighted functions. For instance, Zhang *et al.* [2] used local polynomial methods to estimate nonparametric part of the linear part of the semivarying coefficient model. Huang *et al.* [7] used polynomial splines to estimate nonparametric coefficient functions in varying coefficient models. Qingguo [8] used piecewise local polynomial approximation to obtain parametric estimators of the model. This approach maintains the flexibility of the model while accurately estimating the model parameters. Furthermore, Huang and Zhang [9] proposed using B-spline basis functions to approximate the varying coefficient part. Feng *et al.* [10] used B-spline basis function to solve partial linear model. In addition, Cai *et al.* [11, 12] and Sun *et al.* [13] used a modified Huber function to make the estimation of partial linear models robust. Li *et al.* [14] developed a kernel-weighted local least squares method to estimate smooth coefficient functions, proving the consistency and asymptotic normality of the estimators.

It is critical to perform variable selection in high-dimensional data analysis. Effective variable selection not only simplifies models and reduces computational costs but also enhances the model's generalization ability and prediction accuracy. In recent years, the penalty function, as a variable selection tool, has received widespread attention and has become a hot research topic. This method adds penalty terms to the loss function to identify the most critical variables, improving model interpretability and predictive ability.

For example, Tibshirani [15] proposed the LASSO (Least Absolute Shrinkage and Selection Operator) penalty method to perform variable selection and parameter estimation simultaneously within a model. Zou *et al.* [16] proposed the adaptive LASSO model, which possesses oracle properties. Knight and Fu [17] introduced the more stable Bridge estimation. Tibshirani *et al.* [18] proposed the Fused LASSO method through examples of proteomic and gene expression data. Fan and Li [19] proposed the SCAD (Smoothly Clipped Absolute Deviation) penalty based on the least squares, which can effectively reduce model complexity. Zou *et al.* [20] proposed the elastic net regularization model by combining the LASSO and ridge regression models. Zhao *et al.* [21] studied the high-dimensional partially linear varying coefficient model based on elastic net. In addition, various other variable selection methods have been proposed, such as the group LASSO discussed by Yuan and Lin [22], Friedman *et al.* [23], and Zhou *et al.* [24]. Petry *et al.* [25] and She [26] considered clustered LASSO.

When dealing with high-dimensional data, Zou *et al.* [27] proposed a linear regression model with adaptive elastic net. On one hand, it personalizes the shrinkage of variable coefficients to avoid over-compression; On the other hand, it reasonably allocates collinear variable coefficients to effectively handle collinearity problems. Compared to the elastic net method, the adaptive elastic net retains key variables and eliminates less influential ones through differential weighting. In addition, the adaptive elastic net possesses properties such as asymptotic normality, sparsity, weak consistency, and group effect.

In this paper, we introduce an adaptive elastic net regularization term in the partially linear varying coefficient model and use the ADMM (Alternating Direction Method of Multipliers) in [28] to estimate the parameters. The ADMM proposed by Glowinski in the 1970s, effectively balances decomposability and convergence speed through its alternating optimization of variables and coordination of multiplier updates. It provides new framework and methods for variable selection in partially linear varying coefficient models for high-dimensional data.

The rest of this paper is organized as follows. In Section 2, we construct a variable selection model based on the adaptive elastic net. In Section 3, we propose an ADMM approach to solve the adaptive elastic net model. In Section 4, we demonstrate the convergence of the algorithm. In Section 5, we evaluate the performance of the proposed method through numerical experiments with high-dimensional data. In Section 6, we present an empirical application. Finally, some conclusions are made in Section 7.

2. Construction of variable selection model based on adaptive elastic net

Consider the independent and identically distributed samples $(Y_i, Z_i, T_i), i = 1, \cdots, n$, the partially linear varying coefficient model is as follows:

$$Y_i = X_i^{\mathrm{T}} \beta + Z_i^{\mathrm{T}} \alpha(T_i) + \varepsilon_i, i = 1, 2, \ldots, n, \tag{2.1}$$

where Y_i is the response variable, X_i is a $p \times 1$ covariate, $X_i = (X_{i1}, X_{i2}, \cdots, X_{ip})^T$, β is a $p \times 1$ unknown parameter vector, $\beta = (\beta_1, \beta_2, \cdots, \beta_p)^T$, Z_i is the covariate, $Z_i = (Z_{i1}, Z_{i2}, \cdots, Z_{iq})^T$, $\alpha(\cdot)$ is a $q \times 1$ vector of unknown coefficient functions, $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_q)^T$, ε_i is the random error, and $\varepsilon_i \sim N(0, \sigma^2)$.

Let $Y = (Y_1, Y_2, \cdots, Y_n)^T$, $X = (X_1, X_2, \cdots, X_n)^T$, $Z = (Z_1, Z_2, \cdots, Z_n)^T$, $\varepsilon = (\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n)^T$and $M = \left(Z_1^{\mathrm{T}} \alpha(T_1), Z_2^{\mathrm{T}} \alpha(T_2), \cdots, Z_n^{\mathrm{T}} \alpha(T_n)\right)^{\mathrm{T}}$. Then the model (2.1) can be expressed as:

$$Y = X\beta + M + \varepsilon. \tag{2.2}$$

When X is observable and β is known, model (2.1) can be written as:

$$Y_i^* = Z_i^T \alpha(T_i) + \varepsilon_i, \quad i = 1, 2, \cdots, n, \tag{2.3}$$

where $Y_i^* = Y_i - X_i^T \beta$. To estimate the function coefficient $\alpha_j(t)$, we adopt the local polynomial method as described in [29]. When t is in the neighborhood of t_0, we use the Taylor expansion for $\alpha_j(t)$ and obtain:

$$\alpha_j(t) \approx \alpha_j(t_0) + \alpha'_j(t_0)(t - t_0) = c_j + d_j(t - t_0), j = 1, 2, \ldots, q.$$

Substitute it into model (2.1), thus transforming the problem of solving $\alpha_j(t)$ into a problem of solving unknown parameter c_j, d_j . A local least squares problem is adopted for estimation and solution, then

$$\min_{c,d} \sum_{i=1}^{n} \left[Y_i^* - Z_i^{\mathrm{T}}(c + d(T_i - t_0))\right]^2 K_h(T_i - t_0),$$

where $c = (c_1, c_2, \cdots, c_q)^{\mathrm{T}}, d = (d_1, d_2, \cdots, d_q)^{\mathrm{T}}$, $K_h(\cdot) = K(\cdot/h)/h$ is the kernel function, and h is the bandwidth.

Regarding the minimization of equations c and d, we obtain

$$\left\{\hat{c}_1(t), \cdots \hat{c}_q(t), h\hat{d}_1(t), \cdots, h\hat{d}_q(t)\right\}^{\mathrm{T}} = \left[D_t^{\mathrm{T}} \Omega_t D_t\right]^{-1} D_t^{\mathrm{T}} \Omega_t (Y - X\beta),$$

where

$$D_t = \begin{pmatrix} Z_1^{\mathrm{T}} & \frac{T_1 - t}{h} Z_1^{\mathrm{T}} \\ \vdots & \vdots \\ Z_n^{\mathrm{T}} & \frac{T_n - t}{h} Z_n^{\mathrm{T}} \end{pmatrix},$$

$$\Omega_t = diag\left(K_h\left(T_1 - t\right), \cdots, K_h\left(T_n - t\right)\right).$$

By substituting the estimated function coefficients into (2.2), we obtain the nonparametric component as follows:

$$\begin{aligned} \hat{M} &= \left(Z_1^{\mathrm{T}} \hat{\alpha}\left(T_1\right), Z_2^{\mathrm{T}} \hat{\alpha}\left(T_2\right), \cdots, Z_n^{\mathrm{T}} \hat{\alpha}\left(T_n\right)\right)^{\mathrm{T}} \\ &= \begin{pmatrix} \left(Z_1^{\mathrm{T}} : 0\right)\left[D_{t_1}^{\mathrm{T}} \Omega_{t_1} D_{t_1}\right]^{-1} D_{t_1}^{\mathrm{T}} \Omega_{t_1} \\ \vdots \\ \left(Z_n^{\mathrm{T}} : 0\right)\left[D_{t_n}^{\mathrm{T}} \Omega_{t_n} D_{t_n}\right]^{-1} D_{t_n}^{\mathrm{T}} \Omega_{t_n} \end{pmatrix} (Y - X\beta). \end{aligned}$$

Let

$$S = \begin{pmatrix} \left(Z_1^{\mathrm{T}} : 0\right)\left[D_{t_1}^{\mathrm{T}} \Omega_{t_1} D_{t_1}\right]^{-1} D_{t_1}^{\mathrm{T}} \Omega_{t_1} \\ \vdots \\ \left(Z_n^{\mathrm{T}} : 0\right)\left[D_{t_n}^{\mathrm{T}} \Omega_{t_n} D_{t_n}\right]^{-1} D_{t_n}^{\mathrm{T}} \Omega_{t_n} \end{pmatrix},$$

then the nonparametric component estimation of the partially linear varying coefficient model can be expressed as:

$$\hat{M} = S\left(Y - X\beta\right).$$

By substituting the nonparametric estimate $\hat{M}$ into model (2.2), we can obtain the following equation:

$$Y = X\beta + S\left(Y - X\beta\right) + \varepsilon.$$

By transforming the above equation, we obtain:

$$\left(I_n - S\right) Y = \left(I_n - S\right) X\beta + \varepsilon,$$

so we can transform the model (2.2) into the following linear model:

$$\tilde{Y} = \tilde{X}\beta + \varepsilon,$$

where $\tilde{Y} = (I - S)Y$, $\tilde{X} = (I - S)X$. The loss function is defined by [21] based on the least squares criterion and introduces the elastic net penalty to estimate the parameter β, and constructs the following elastic net estimator:

$$\min_{\beta} \frac{1}{2}\left\|\tilde{Y} - \tilde{X}\beta\right\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2, \tag{2.4}$$

where λ_1 and λ_2 are nonnegative regularization parameters.

When selecting variable features, the elastic net considers variables according to the same weight. In practice, some variables play an important role in feature selection, while some variables play a weaker role, so we consider giving feature weights. Based on (2.4), we add weight to ℓ_1 penalty and propose the following adaptive elastic net estimator:

$$\min_{\beta} \frac{1}{2}\left\|\tilde{Y}-\tilde{X}\beta\right\|_2^2+\lambda_1^*\sum_{j=1}^{p}\omega_j|\beta_j|+\lambda_2\|\beta\|_2^2, \tag{2.5}$$

where $\{\omega_j\}_{j=1}^p$ are the weights, λ_1^* and λ_2 are nonnegative regularization parameters. λ_1^* and λ_1 are different values.

Generally speaking, the weight selection of the adaptive elastic net is determined based on the estimates of the general elastic net, which denotes $\hat{\beta}_{(enet)}$:

$$\hat{\beta}_{(enet)}=\arg\min_{\beta}\left\{\frac{1}{2}\left\|\tilde{Y}-\tilde{X}\beta\right\|_2^2+\lambda_1\left\|\beta\right\|_1+\lambda_2\left\|\beta\right\|_2^2\right\}.$$

We construct the adaptive weights by

$$\omega_j=\left|\hat{\beta}_{j(enet)}\right|^{-\gamma},\ j=1,2,...,p,$$

where $\gamma>0$.

However, the coefficients estimated through elastic net may be 0, so the weight ω_j is defined in this paper as:

$$\omega_j=\left(\left|\hat{\beta}_{j(enet)}\right|+\frac{1}{n}\right)^{-\gamma},$$

where $\gamma>0$, $\gamma=1$, n is the sample size.

By introducing the auxiliary variable α, problem (2.5) is transformed into the following optimization problem:

$$\begin{aligned}&\min_{\beta} \frac{1}{2}\left\|\tilde{Y}-\tilde{X}\beta\right\|_2^2+\lambda_1^*\sum_{j=1}^{p}\omega_j|\alpha_j|+\lambda_2\|\alpha\|_2^2\\&\text{s.t.}\quad \beta-\alpha=0.\end{aligned} \tag{2.6}$$

Compared to the elastic net, adaptive elastic net improves the impact of variables on the model by handling them with differential weights. That is, the partially linear varying coefficient model with adaptive elastic net (abbreviated as AENVCPLM), which transforms the parameter solving problem into solving the solution of a constrained optimization problem.

3. Solving the adaptive elastic net variable selection model with ADMM

In this section, we will elaborate on how to use the ADMM to solve AENVCPLM. We first construct an augmented Lagrangian function based on the optimization problem (2.6), and then splits a complex optimization problem into multiple more easily solvable subproblems. Finally, we solve these subproblems one by one, integrate the results, and obtain the solution to the original optimization problem.

For problem (2.6), its augmented Lagrangian function is as follows:

$$L_\rho(\beta,\alpha,u)=\frac{1}{2}\left\|\tilde{Y}-\tilde{X}\beta\right\|_2^2+\lambda_1^*\sum_{j=1}^{p}\omega_j|\alpha_j|+\lambda_2\|\alpha\|_2^2-\langle u,\beta-\alpha\rangle+\frac{\rho}{2}\|\beta-\alpha\|_2^2,$$

where u is the Lagrange multiplier and ρ is the penalty parameter.

Utilizing the ADMM algorithm in [28], the $k+1$-th iteration step for problem (2.6) is as follows:

$$\begin{cases}\beta^{k+1}=\arg\min\limits_{\beta} L_\rho(\beta,\alpha^k,u^k),\\ \alpha^{k+1}=\arg\min\limits_{\alpha} L_\rho(\beta^{k+1},\alpha,u^k),\\ u^{k+1}=u^k-\rho(\beta^{k+1}-\alpha^{k+1}),\end{cases}$$

where $\left(\beta^k,\alpha^k,u^k\right)$ denotes the k-th iteration step.

By calculation, the β-subproblem can be written as:

$$\beta^{k+1}=\arg\min_{\beta}\left\{\frac{1}{2}\left\|\tilde{Y}-\tilde{X}\beta\right\|_2^2-\langle u^k,\beta-\alpha^k\rangle+\frac{\rho}{2}\|\beta-\alpha^k\|_2^2\right\}.$$

According to the aforementioned method, we can find the partial derivative for any given β :

$$\frac{\partial L_\rho(\beta,\alpha^k,u^k)}{\partial\beta}=-\tilde{X}^T\tilde{Y}+\tilde{X}^T\tilde{X}\beta-(u^T)^k+\rho(\beta-\alpha^k)=0.$$

The analytical solution for β can be expressed in the following form:

$$\beta^{k+1}=\left(\tilde{X}^T\tilde{X}+\rho I\right)^{-1}\left(\tilde{X}^T\tilde{Y}+(u^T)^k+\rho I\alpha^k\right). \tag{3.1}$$

In the α-subproblem, due to the presence of $\lambda_1^*\sum_{j=1}^{p}\omega_j|\alpha_j|$, it does not have a closed-form solution. We can consider the following soft-thresholding method to obtain a closed-form solution for this subproblem:

$$x=\arg\min_{x}\left\{t\|x\|_1+\frac{r}{2}\|x-k\|_2^2\right\},$$

the closed-form solution for this equation can be represented as:

$$x:=S_{t/r}(k):=\operatorname{sign}(k)\cdot\max\left\{|k|-\frac{t}{r},0\right\},$$

where r is a scalar.

Thus, the solution to the α-subproblem can be expressed as:

$$\begin{aligned}\alpha^{k+1} &= \arg\min_{\alpha}\left\{\lambda_1^*\sum_{j=1}^{p}\omega_j|\alpha_j| + \lambda_2\|\alpha\|_2^2 - \langle u^k, \beta^{k+1}-\alpha\rangle + \frac{\rho}{2}\|\beta^{k+1}-\alpha\|_2^2\right\}\\ &= \arg\min_{\alpha}\left\{\lambda_1^*\sum_{j=1}^{p}\omega_j|\alpha_j| + \frac{\rho+2\lambda_2}{2}\left\|\alpha-\left(\beta^{k+1}+\frac{u^k}{\rho+2\lambda_2}\right)\right\|_2^2\right\}\\ &= S_{\lambda_1^*\omega/\rho+2\lambda_2}\left(\beta^{k+1}+\frac{u^k}{\rho+2\lambda_2}\right),\end{aligned} \tag{3.2}$$

where ω is the weight vector.

Combining the above analysis, the solution to the optimization problem (2.6) can be represented by the following iterative framework:

$$\begin{cases}\beta^{k+1} = \left(\tilde{X}^T\tilde{X}+\rho I\right)^{-1}\left(\tilde{X}^T\tilde{Y}+(u^T)^k+\rho I\alpha^k\right),\\ \alpha^{k+1} = S_{\lambda_1^*\omega/\rho+2\lambda_2}\left(\beta^{k+1}+\dfrac{u^k}{\rho+2\lambda_2}\right),\\ u^{k+1} = u^k-\rho(\beta^{k+1}-\alpha^{k+1}).\end{cases} \tag{3.3}$$

The iterative process for solving (2.6) with ADMM is as follows:

Algorithm: ADMM Iterative Framework for Solving AENVCPLM.

Input: X, Z, and Y. Give the initial variable (β^0, α^0, u^0), and set the penalty parameters $\lambda_1^* > 0, \lambda_2 > 0, \rho > 0$.
1: Set the iteration step $k = 1, 2, \ldots, n$;
2: If the stopping criterion is met, then stop the iteration; otherwise, set $k = k+1$, go to 3;
3: Calculate β^{k+1} using equation (3.1);
4: Calculate α^{k+1} using equation (3.2);
5: Calculate u^{k+1} : $u^{k+1} = u^k - \rho(\beta^{k+1} - \alpha^{k+1})$;
6: end
Output: (β^l, α^l, u^l), where l is the number of iterations.

4. Convergence analysis

In this section, we will analyze the convergence of the algorithm . The augmented Lagrangian function for the optimization problem (2.6) is given by:

$$L_\rho(\beta,\alpha,u) = \frac{1}{2}\|\tilde{Y}-\tilde{X}\beta\|_2^2 + \lambda_1^*\sum_{j=1}^{p}\omega_j|\alpha_j| + \lambda_2\|\alpha\|_2^2 - \langle u, \beta-\alpha\rangle + \frac{\rho}{2}\|\beta-\alpha\|_2^2,$$

where u is the lagrange multiplier. According to the first-order optimality conditions, we obtain:

$$\begin{cases} \theta_1(\beta)-\theta_1(\beta^{k+1})+(\beta-\beta^{k+1})^T\{-I^Tu^{k+1}+\rho I^T(\beta^{k+1}-\alpha^k)\}\geq 0,\\ \theta_2(\alpha)-\theta_2(\alpha^{k+1})+(\alpha-\alpha^{k+1})^T\{Iu^k-\rho I(\beta^{k+1}-\alpha^{k+1})\}\geq 0,\\ (u-u^{k+1})^T(\beta^{k+1}-\alpha^{k+1})+\dfrac{1}{\rho}(u^{k+1}-u^k)\geq 0. \end{cases} \tag{4.1}$$

where $\theta_1(\beta)=\frac{1}{2}\|\tilde{Y}-\tilde{X}\beta\|_2^2$, $\theta_2(\alpha)=\lambda_2\|\alpha\|_2^2+\lambda_1^*\sum_{j=1}^{p}\omega_j\,|\alpha_j|$. ϖ is the solution set satisfying equation (4.1).

Let

$$b=\begin{pmatrix}\beta\\ \alpha\end{pmatrix},\psi=\begin{pmatrix}\beta\\ \alpha\\ u\end{pmatrix},F(\psi)=\begin{pmatrix}-I^Tu\\ Iu\\ \beta-\alpha\end{pmatrix},$$

then

$$\theta(b)=\theta_1(\beta)+\theta_2(\alpha),$$

equation (4.1) is transformed into the following variational inequality:

$$\theta(b)-\theta(b^*)+(\psi-\psi^*)^TF(\psi^*)\geq 0, \tag{4.2}$$

where $\psi^*\in\varpi$, ϖ^* is the solution set that satisfies equation (4.2). For convenience, we let

$$v=\begin{pmatrix}\alpha\\ u\end{pmatrix},V^*=\{(\alpha^*,u^*)\mid(\beta^*,\alpha^*,u^*)\in\varpi^*\}.$$

Now, by substituting $u^{k+1}=u^k-\rho(\beta^{k+1}-\alpha^{k+1})$ into equation (4.1), and eliminating u^k, we obtain the following equation:

$$\begin{cases} \theta_1(\beta)-\theta_1(\beta^{k+1})+(\beta-\beta^{k+1})^T\{-I^Tu^{k+1}-\rho I^TI(\alpha^k-\alpha^{k+1})\}\geq 0,\\ \theta_2(\alpha)-\theta_2(\alpha^{k+1})+(\alpha-\alpha^{k+1})^T(Iu^{k+1})\geq 0. \end{cases} \tag{4.3}$$

For convenience, let $b^{k+1}=(\beta^{k+1},\alpha^{k+1})$ then equation (4.3) can be written as:

$$\begin{aligned} &\theta(b)-\theta(b^{k+1})+\begin{pmatrix}\beta-\beta^{k+1}\\ \alpha-\alpha^{k+1}\end{pmatrix}^T\left\{\begin{pmatrix}-I^Tu^{k+1}\\ I^Tu^{k+1}\end{pmatrix}\right.\\ &\left.-\rho\begin{pmatrix}I^T\\ -I^T\end{pmatrix}I(\alpha^k-\alpha^{k+1})+\begin{pmatrix}0&0\\ 0&\rho I^TI\end{pmatrix}\begin{pmatrix}\beta^{k+1}-\beta^k\\ \alpha^{k+1}-\alpha^k\end{pmatrix}\right\}\geq 0, \end{aligned} \tag{4.4}$$

combining equation (4.4) with equation (4.1), we have $\psi^{k+1} \in \varpi$, and for any $\psi \in \varpi$, the following equation holds:

$$\begin{aligned}\theta(b) - \theta(b^{k+1}) + \begin{pmatrix} \beta - \beta^{k+1} \\ \alpha - \alpha^{k+1} \\ u - u^{k+1} \end{pmatrix}^T \Bigg\{ \begin{pmatrix} -I^T u^{k+1} \\ I^T u^{k+1} \\ \beta^{k+1} - \alpha^{k+1} \end{pmatrix} \\ + \rho \begin{pmatrix} I^T \\ -I^T \\ 0 \end{pmatrix} (-I)(\alpha^k - \alpha^{k+1}) + \begin{pmatrix} 0 & 0 \\ \rho I^T I & 0 \\ 0 & \frac{1}{\rho} I_n \end{pmatrix} \begin{pmatrix} \alpha^{k+1} - \alpha^k \\ u^{k+1} - u^k \end{pmatrix} \Bigg\} \geq 0.\end{aligned} \tag{4.5}$$

Since $\psi^{k+1} \in \varpi$, for any $\psi \in \varpi$, the above equation can be rewritten as:

$$\begin{aligned}\theta(b) - \theta(b^{k+1}) + (\psi - \psi^{k+1}) F(\psi^{k+1}) + \rho \begin{pmatrix} \beta - \beta^{k+1} \\ \alpha - \alpha^{k+1} \\ u - u^{k+1} \end{pmatrix} \begin{pmatrix} I^T \\ -I^T \\ 0 \end{pmatrix} (-I)(\alpha^k - \alpha^{k+1}) \\ \geq \begin{pmatrix} \alpha - \alpha^{k+1} \\ u - u^{k+1} \end{pmatrix}^T \begin{pmatrix} \rho I^T I & 0 \\ 0 & \frac{1}{\rho} I_n \end{pmatrix} \begin{pmatrix} \alpha^k - \alpha^{k+1} \\ u^k - u^{k+1} \end{pmatrix}.\end{aligned} \tag{4.6}$$

The convergence of the algorithm proposed in this paper is given by the following theorems.

Theorem 4.1. *Let $\psi^{k+1} = (\beta^{k+1}, \alpha^{k+1}, u^{k+1})$ be the sequence generated by the algorithm, then we have:*

$$(v^{k+1} - v^*)^T H (v^k - v^{k+1}) \geq (\psi^{k+1} - \psi^*)^T \eta(\alpha^k, \alpha^{k+1}), \tag{4.7}$$

where

$$\eta(\alpha^k, \alpha^{k+1}) = \rho \begin{pmatrix} I^T \\ -I^T \\ 0 \end{pmatrix} (-I)(\alpha^k - \alpha^{k+1}), H = \begin{pmatrix} \rho I^T I & 0 \\ 0 & \frac{1}{\rho} I_n \end{pmatrix}.$$

Proof. Let $\psi = \psi^*$ in equation (4.6), using H and $\eta(\alpha^k, \alpha^{k+1})$, we obtain:

$$(v^{k+1} - v^*)^T H (v^k - v^{k+1}) \geq (\psi^{k+1} - \psi^*)^T \eta(\alpha^k, \alpha^{k+1}) + \theta(b^{k+1}) - \theta(b^*) + (\psi^{k+1} - \psi^*)^T F(\psi^{k+1}), \tag{4.8}$$

since F is monotonic, it follows that

$$\theta(b^{k+1}) - \theta(b^*) + (\psi^{k+1} - \psi^*)^T F(\psi^{k+1}) \geq \theta(b^{k+1}) - \theta(b^*) + (\psi^{k+1} - \psi^*)^T F(\psi^*) \geq 0.$$

Substituting it in (4.8), the theorem is proved. □

Theorem 4.2. *Let $\psi^{k+1} = (\beta^{k+1}, \alpha^{k+1}, u^{k+1})$ be the sequence generated by the algorithm, then we have:*

$$(\psi^{k+1} - \psi^*)^T \eta(\alpha^k, \alpha^{k+1}) = (u^k - u^{k+1})^T (-I)(\alpha^k - \alpha^{k+1}), \tag{4.9}$$

and

$$(u^k - u^{k+1})^T (-I)(\alpha^k - \alpha^{k+1}) \geq 0. \tag{4.10}$$

Proof. From Theorem 4.1, we have $\eta(\alpha^k, \alpha^{k+1})$, $\beta^* - \alpha^* = 0$, and

$$u^{k+1} = u^k - \rho(\beta^{k+1} - \alpha^{k+1}).$$

We obtain,

$$\begin{aligned}(\psi^{k+1} - \psi^*)^T \eta(\alpha^k, \alpha^{k+1}) &= (-I(\alpha^k - \alpha^{k+1}))^T \rho \left\{(\beta^{k+1} - \alpha^{k+1}) - (\beta^* - \alpha^*)\right\} \\ &= (-I(\alpha^k - \alpha^{k+1}))^T \rho(\beta^{k+1} - \alpha^{k+1}) \\ &= (u^k - u^{k+1})^T(-I)(\alpha^k - \alpha^{k+1}).\end{aligned}$$

Since equation (4.3) holds for the k-th iteration and the previous iterations, we have:

$$\theta_2(\alpha) - \theta_2(\alpha^{k+1}) + (\alpha - \alpha^{k+1})^T(Iu^{k+1}) \geq 0, \tag{4.11}$$

$$\theta_2(\alpha) - \theta_2(\alpha^k) + (\alpha - \alpha^k)^T(Iu^{k+1}) \geq 0. \tag{4.12}$$

Set $\alpha = \alpha^k$ in equation (4.11) and $\alpha = \alpha^{k+1}$ in equation (4.12), and add the resulting two new equation to obtain:

$$(u^k - u^{k+1})^T(-I)(\alpha^k - \alpha^{k+1}) \geq 0,$$

which completes the proof. □

Theorem 4.3. *Let $\psi^{k+1} = (\beta^{k+1}, \alpha^{k+1}, u^{k+1})$ be the sequence generated by the algorithm. For $\forall v^* \in V^*$, we have:*

$$(v^{k+1} - v^*)^T H(v^k - v^{k+1}) \geq 0. \tag{4.13}$$

Proof. This can be directly proven from (4.7), (4.9), and (4.10). □

Remark 4.4. In this paper, H is positive definite, $\|v - \tilde{v}\|_H$ is represented as:

$$\|v - \tilde{v}\|_H^2 = (v - \tilde{v})^T H(v - \tilde{v}) = \rho\| - I(\alpha - \tilde{\alpha})\|^2 + \frac{1}{\rho}\|u - \tilde{u}\|^2.$$

Theorem 4.5. *Let $\psi^{k+1} = (\beta^{k+1}, \alpha^{k+1}, u^{k+1})$ be the sequence generated by the algorithm. For $\forall v^* \in V^*$, we have:*

$$\|v^{k+1} - v^*\|_H^2 \leq \|v^k - v^*\|_H^2 - \|v^k - v^{k+1}\|_H^2. \tag{4.14}$$

Proof. From (4.13), we have:

$$\begin{aligned}\|v^k - v^*\|_H^2 &= \|(v^{k+1} - v^*) + (v^k - v^{k+1})\|_H^2 \\ &= \|v^{k+1} - v^*\|_H^2 + 2(v^{k+1} - v^*)^T H(v^k - v^{k+1}) + \|v^k - v^{k+1}\|_H^2 \\ &\geq \|v^{k+1} - v^*\|_H^2 + \|v^k - v^{k+1}\|_H^2,\end{aligned}$$

which completes the proof. □

So far, we have fully proved Theorems 4.1–4.5. Based on equation (4.14), the following corollary obviously holds.

Corollary 4.6. *Based on the sequence generated by the algorithm, we can draw the following conclusions:*

1. $\lim_{k\to\infty} ||v^k - v^{k+1}||_H = 0$;
2. *The sequence* $\{\|v^k - v^{k+1}\|_H\}$ *is non-increasing, for* $\forall v^* \in \varpi^*$;
3. *The sequence* $\{v^k\}$ *is bounded.*

Through the corollary, we have proven that v^k converges to v^*. Therefore, the convergence of the algorithm has been proven.

5. Numerical experiments

In this section, we demonstrate the effectiveness of the partially linear varying coefficient model with an adaptive elastic net penalty term through numerical experiments. We compare it with other modeling methods such as elastic net, group lasso, and SCAD penalty terms. All models are fitted using the ADMM algorithm in high-dimensional settings. We will compare the performance of four methods in terms of mean square error and relative error. Numerical experiments are completed in the Windows 11 operating system and MATLAB R2022a software.

5.1. Data setting and parameter selection

In the experiments, the data follows the characteristics of model (2.2). The sample size is $n = 100$, and the covariate X follows a normal distribution with mean zero and covariance matrix Σ, $X \sim N(0, \Sigma)$. The i-th and j-th elements of Σ have a covariance of $\Sigma = 0.5^{|j-i|}$. β is a p-dimensional vector, where $\beta_1 = 1, \beta_2 = 2, \beta_3 = 0.5, \beta_4 = -1$, and other elements are 0. In all simulation experiments, we mainly consider the case of varying coefficient part $\alpha(T) \in \mathrm{R}^3$ in the partially linear varying coefficient model.

Let $\alpha_1(T) = \sin(8\pi T), \alpha_2(T) = \cos(3\pi T), \alpha_3(T) = (T - 0.2)^2$. Here the variable T is uniformly distributed over $[0, 1]$, $T \sim U[0, 1]$. The random error is $\varepsilon \sim N(0, \sigma^2)$, where σ takes values 0.5, 1, 2. Correspondingly, we set $Z_1 \sim N(0, 1)$, $Z_2 \sim N(0, 1.5)$, $Z_3 \sim N(0, 2)$. Additionally, we employ the local polynomial method to estimate the varying coefficient component in the model. The bandwidth is crucial for the performance of the local polynomial regression model. The bandwidth is set as $h = \frac{1}{2}n^{-\frac{1}{5}}$, and the kernel function is the Gaussian kernel $K(u) = \frac{1}{\sqrt{2\pi}}\mathrm{e}^{-\frac{1}{2}u^2}$.

In experiments, we use the cross-validation (CV) method to select parameters. The parameter selection for the adaptive elastic net model is: $\lambda_1^* = 0.06, \lambda_2 = 0.1, \rho = 0.55$; The parameter selection for the elastic net model is: $\lambda_1 = 0.55, \lambda_2 = 0.1, \rho = 0.55$; The parameter selection for the group lasso model is: $\lambda = 0.99, \rho = 0.55$; The parameter selection for the SCAD model is: $\lambda = 0.005, \rho = 0.55$. The sample size of the experiments is 100, the feature dimensions are different.

5.2. Results and analysis

Simulation experiments were conducted for four partially linear varying coefficient models. These models were evaluated using the high-dimensional data and parameters generated by the methods above. It is mainly compared from two aspects: mean square error and relative error. The expressions for MSE and RE are as follows:

$$MSE = \frac{1}{p}\|\hat{y} - y^*\|^2,$$
$$RE = \frac{\|\hat{y} - y^*\|^2}{\|y^*\|^2},$$

where $\hat{y}$ is the estimated value, y^* is the true value, and p is the dimension.

The main results are shown in Table 1–Table 3. In the tables, AEN, EN, GL, and SCAD respectively represent the the adaptive elastic net partially linear varying coefficient model, the elastic net partially linear

TABLE 1. Mean squared errors processed by four methods.

(n,p)	Method	MSE $\sigma=0.5$	MSE $\sigma=1$	MSE $\sigma=2$
(100,40)	AEN	$4.4494e-04$	$3.4000e-03$	$3.2600e-02$
	EN	$3.8000e-03$	$1.7500e-02$	$7.5800e-02$
	GL	$1.7990e-02$	$3.3250e-02$	$9.4760e-02$
	SCAD	$2.6650e-02$	$4.2720e-02$	$1.1460e-01$
(100,60)	AEN	$4.9881e-04$	$2.5000e-03$	$1.8200e-02$
	EN	$3.0000e-03$	$1.6200e-02$	$7.8000e-02$
	GL	$5.2500e-02$	$7.9370e-02$	$1.7040e-01$
	SCAD	$9.4040e-02$	$1.2790e-01$	$2.3710e-01$
(100,80)	AEN	$2.1180e-04$	$2.6000e-03$	$2.9500e-02$
	EN	$6.5000e-03$	$4.0100e-02$	$2.3380e-01$
	GL	$6.1500e-02$	$1.0860e-01$	$3.5910e-01$
	SCAD	$2.0970e-01$	$3.7550e-01$	$4.2420e-01$
(100,100)	AEN	$3.1936e-04$	$2.5000e-03$	$2.3700e-02$
	EN	$2.9000e-03$	$1.8800e-02$	$1.1160e-01$
	GL	$5.7530e-02$	$8.1600e-02$	$1.9280e-01$
	SCAD	$2.4570e-01$	$4.0860e-01$	$8.4100e-01$
(100,260)	AEN	$1.3883e-04$	$1.3000e-03$	$9.5000e-03$
	EN	$1.3000e-03$	$5.0000e-03$	$1.9100e-02$
	GL	$5.8630e-03$	$7.9380e-03$	$1.7270e-02$
	SCAD	$1.7950e-02$	$1.9380e-02$	$2.5940e-02$
(100,560)	AEN	$5.4802e-05$	$1.0427e-04$	$2.3293e-04$
	EN	$8.7124e-04$	$2.1000e-03$	$6.1000e-03$
	GL	$2.8380e-03$	$3.4220e-03$	$5.7290e-03$
	SCAD	$9.8350e-03$	$1.0060e-02$	$1.1110e-02$
(100,860)	AEN	$4.1123e-05$	$1.2937e-04$	$3.4291e-04$
	EN	$5.5500e-04$	$1.6000e-03$	$3.6000e-03$
	GL	$2.0720e-03$	$2.4270e-03$	$3.6560e-03$
	SCAD	$6.6850e-03$	$6.7710e-03$	$7.1160e-03$
(100,1160)	AEN	$6.0636e-05$	$2.2629e-04$	$8.5229e-04$
	EN	$6.8586e-04$	$1.6000e-03$	$3.5000e-03$
	GL	$3.0580e-03$	$3.3130e-03$	$4.0540e-03$
	SCAD	$5.2530e-03$	$5.2770e-03$	$5.4580e-03$
(100,1460)	AEN	$2.0172e-05$	$7.5773e-05$	$2.6402e-04$
	EN	$3.2748e-04$	$9.8269e-04$	$2.2000e-03$
	GL	$2.1970e-03$	$2.2810e-03$	$2.7060e-03$
	SCAD	$4.2270e-03$	$4.2410e-03$	$4.3220e-03$
(100,1760)	AEN	$3.8386e-05$	$1.4044e-04$	$5.4518e-04$
	EN	$5.9848e-04$	$1.2000e-03$	$2.5000e-03$
	GL	$2.3640e-03$	$2.6110e-03$	$3.2670e-03$
	SCAD	$3.6980e-03$	$3.6920e-03$	$3.7360e-03$
(100,2060)	AEN	$1.0948e-05$	$4.3061e-05$	$1.6513e-04$
	EN	$3.1279e-04$	$8.3860e-04$	$1.4000e-03$
	GL	$1.9170e-03$	$2.0230e-03$	$2.3270e-03$
	SCAD	$3.3460e-03$	$3.3020e-03$	$3.1590e-03$
(100,2360)	AEN	$1.7334e-05$	$6.8091e-05$	$2.6659e-04$
	EN	$3.9385e-04$	$8.1717e-04$	$1.2000e-03$
	GL	$1.8900e-03$	$1.9340e-03$	$2.1300e-03$
	SCAD	$3.0230e-03$	$2.9830e-03$	$2.8190e-03$
(100,2660)	AEN	$2.2187e-04$	$3.2973e-05$	$1.3850e-04$
	EN	$3.0054e-04$	$6.3114e-04$	$1.3000e-03$
	GL	$1.5070e-03$	$1.5770e-03$	$1.7970e-03$
	SCAD	$2.8320e-03$	$2.8220e-03$	$2.7070e-03$
(100,2960)	AEN	$1.7738e-05$	$5.9903e-05$	$2.2024e-04$
	EN	$3.2190e-04$	$7.8504e-04$	$1.2000e-03$
	GL	$1.6740e-03$	$1.7510e-03$	$1.9190e-03$
	SCAD	$2.5180e-03$	$2.5290e-03$	$2.4450e-03$
(100,3260)	AEN	$2.8537e-06$	$9.6006e-06$	$3.5730e-05$
	EN	$2.7821e-04$	$5.2649e-04$	$8.2952e-04$
	GL	$1.4170e-03$	$1.4530e-03$	$1.5720e-03$
	SCAD	$2.2980e-03$	$2.3190e-03$	$2.3190e-03$

Table 2. Relative errors in parameters processed by four methods.

(n,p)	Method	RE		
		$\sigma = 0.5$	$\sigma = 1$	$\sigma = 2$
(100,40)	AEN	0.0534	0.1486	0.4571
	EN	0.1559	0.3343	0.6967
	GL	0.3393	0.4613	0.7788
	SCAD	0.4129	0.5229	0.8564
(100,60)	AEN	0.0692	0.1512	0.4180
	EN	0.1705	0.3943	0.8700
	GL	0.7099	0.8729	1.2709
	SCAD	0.9501	1.1083	1.5086
(100,80)	AEN	0.0521	0.1811	0.6147
	EN	0.2886	0.7168	1.7300
	GL	0.8873	1.1790	2.1440
	SCAD	1.6385	2.1925	2.3302
(100,100)	AEN	0.0715	0.2009	0.6159
	EN	0.2160	0.5481	1.3360
	GL	0.9594	1.1430	1.7570
	SCAD	1.9826	2.5568	3.6683
(100,260)	AEN	0.0760	0.2369	0.6289
	EN	0.2304	0.4539	0.8911
	GL	0.4939	0.5746	0.8476
	SCAD	0.8641	0.8979	1.0388
(100,560)	AEN	0.0701	0.0907	0.1445
	EN	0.2794	0.4297	0.7387
	GL	0.5043	0.5537	0.7165
	SCAD	0.9387	0.9493	0.9979
(100,860)	AEN	0.0752	0.1334	0.2122
	EN	0.2710	0.4711	0.7072
	GL	0.5340	0.5778	0.7093
	SCAD	0.9590	0.9652	0.9894
(100,1160)	AEN	0.1061	0.2049	0.3977
	EN	0.3568	0.5410	0.8078
	GL	0.7534	0.7842	0.8675
	SCAD	0.8741	0.9860	1.0065
(100,1460)	AEN	0.0686	0.1330	0.2483
	EN	0.2766	0.4791	0.7132
	GL	0.7164	0.7300	0.7950
	SCAD	0.9937	0.9953	1.0048
(100,1760)	AEN	0.1040	0.1989	0.3918
	EN	0.4107	0.5851	0.8468
	GL	0.8158	0.8574	0.9591
	SCAD	1.0205	1.0197	1.0257
(100,2060)	AEN	0.0601	0.1191	0.2333
	EN	0.3211	0.5257	0.6782
	GL	0.7949	0.8166	0.8758
	SCAD	1.0502	1.0432	1.0203
(100,2360)	AEN	0.0809	0.1603	0.3173
	EN	0.3836	0.5555	0.6746
	GL	0.8448	0.8546	0.8967
	SCAD	1.0684	1.0612	1.0318
(100,2660)	AEN	0.0567	0.1185	0.2428
	EN	0.3073	0.5183	0.7426
	GL	0.8009	0.8193	0.8746
	SCAD	1.0979	1.0960	1.0734
(100,2960)	AEN	0.0917	0.1684	0.3230
	EN	0.3905	0.6098	0.7391
	GL	0.8903	0.9107	0.9534
	SCAD	1.0921	1.0944	1.0760
(100,3260)	AEN	0.0386	0.0788	0.1365
	EN	0.3809	0.5240	0.6578
	GL	0.8596	0.8707	0.9057
	SCAD	1.0949	1.0998	1.0999

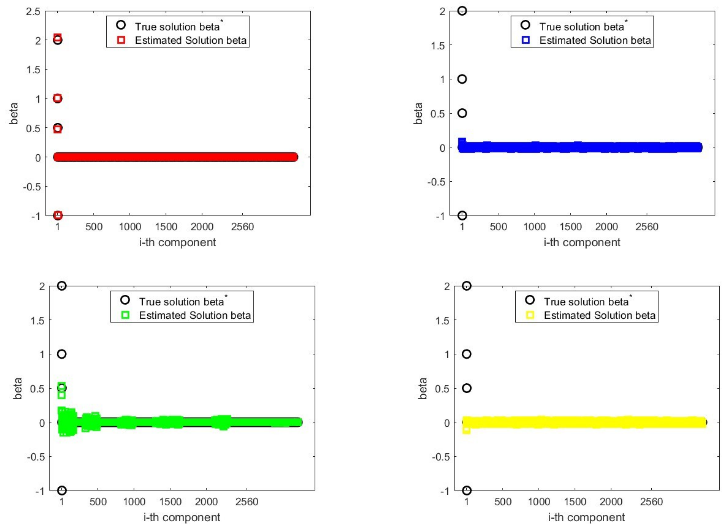

FIGURE 1. Comparison of parameter estimation at $\sigma = 0.5$ and $p = 3260$.

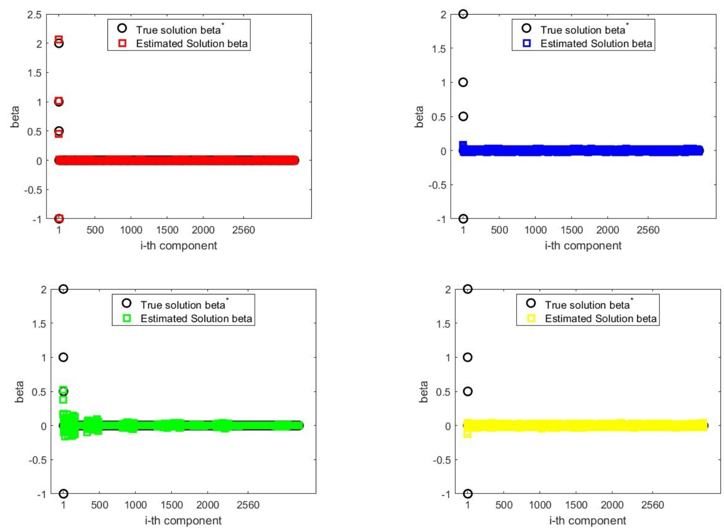

FIGURE 2. Comparison of parameter estimation at $\sigma = 1$ and $p = 3260$.

varying coefficient model, the group lasso partially linear varying coefficient model and the SCAD partially linear varying coefficient model.

The results of Table 1 show that the mean square error of the AEN method is lower than that of the EN, GL, and SCAD methods on high-dimensional data; Even with increasing σ, all methods keep a low mean square error and show good performance. Specifically, the AEN method outperforms the EN, GL, and SCAD methods in handling high-dimensional data.

The results of Table 2 show that the RE of the AEN method is lower than that of the EN, GL and SCAD methods on high-dimensional data. Even if the σ increases, the RE of four methods remains at a low level and their performance is stable. In addition, as the dimensionality of the data increases, the AEN method still maintains significant performance compared to other methods.

TABLE 3. Relative errors in models processed by four methods.

(n, p)	Method	RE		
		$\sigma = 0.5$	$\sigma = 1$	$\sigma = 2$
(100,40)	AEN	0.0064	0.1094	0.2297
	EN	0.2704	0.3649	0.4071
	GL	0.3227	0.4106	0.4653
	SCAD	0.4073	0.4836	0.5100
(100,60)	AEN	0.0325	0.1572	0.2826
	EN	0.3250	0.4555	0.5208
	GL	0.4324	0.4773	0.5290
	SCAD	0.6068	0.6111	0.6042
(100,80)	AEN	0.0492	0.1629	0.3593
	EN	0.4863	0.6819	0.7900
	GL	0.6490	0.6980	0.7768
	SCAD	0.8612	0.8839	0.8928
(100,100)	AEN	0.0583	0.2202	0.4450
	EN	0.5151	0.7323	0.8464
	GL	0.7569	0.8211	0.8858
	SCAD	0.9999	0.9999	0.9982
(100,260)	AEN	0.0972	0.3794	0.6385
	EN	0.6810	0.8672	0.9441
	GL	0.8716	0.9142	0.9577
	SCAD	0.9996	0.9998	0.9999
(100,560)	AEN	0.6586	0.2644	0.6224
	EN	0.6474	0.8690	0.9472
	GL	0.9186	0.9436	0.9706
	SCAD	0.9994	0.9996	0.9998
(100,860)	AEN	1.0214	0.0112	0.6036
	EN	0.6475	0.8591	0.9454
	GL	0.9246	0.9461	0.9723
	SCAD	0.9990	0.9993	0.9997
(100,1160)	AEN	0.9367	0.0690	0.7077
	EN	0.7264	0.8965	0.9600
	GL	0.9469	0.9639	0.9824
	SCAD	0.9990	0.9993	0.9997
(100,1460)	AEN	0.9286	0.0083	0.6978
	EN	0.7140	0.8958	0.9608
	GL	0.9476	0.9654	0.9832
	SCAD	0.9988	0.9992	0.9996
(100,1760)	AEN	0.8321	0.0571	0.7410
	EN	0.7526	0.9099	0.9656
	GL	0.9572	0.9728	0.9869
	SCAD	0.9988	0.9992	0.9996
(100,2060)	AEN	1.4450	0.2505	0.4068
	EN	0.6600	0.8698	0.9518
	GL	0.9461	0.9619	0.9811
	SCAD	0.9983	0.9988	0.9994
(100,2360)	AEN	0.7582	0.1101	0.5643
	EN	0.7677	0.9163	0.9679
	GL	0.9611	0.9740	0.9875
	SCAD	0.9987	0.9991	0.9996
(100,2660)	AEN	0.7216	0.0156	0.4911
	EN	0.7217	0.8985	0.9625
	GL	0.9560	0.9726	0.9873
	SCAD	0.9986	0.9991	0.9996
(100,2960)	AEN	1.0470	0.0493	0.4746
	EN	0.7215	0.8988	0.9628
	GL	0.9654	0.9768	0.9886
	SCAD	0.9990	0.9993	0.9996
(100,3260)	AEN	1.2215	0.1592	0.4163
	EN	0.6899	0.8858	0.9574
	GL	0.9643	0.9740	0.9866
	SCAD	0.9988	0.9992	0.9995

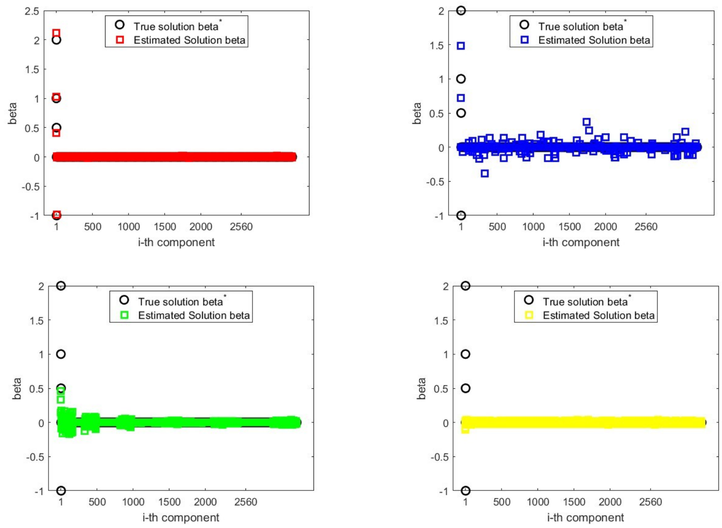

FIGURE 3. Comparison of parameter estimation at $\sigma = 2$ and $p = 3260$.

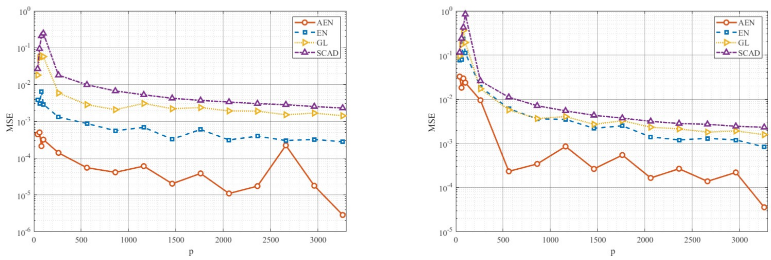

FIGURE 4. Comparison of MSE at $\sigma = 0.5$ (left) and $\sigma = 2$ (right).

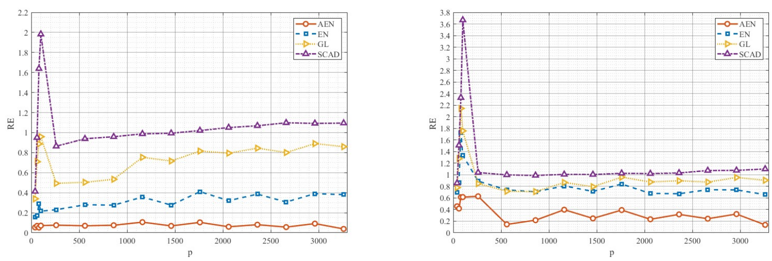

FIGURE 5. Comparison the RE of parameters at $\sigma = 0.5$ (left) and $\sigma = 2$ (right).

The results of Table 3 show that in high-dimensional, high-variance settings, the AEN method keeps a low RE. In low-dimensional, low-variance settings, the EN method has a smaller RE than AEN. Overall, AEN performs better in high dimensions and high variance. EN is more stable in low dimensions and low variance. Under the condition of small variance and high dimension, we can reduce the relative error through the adaptive weight improvement method.

For easier comparison, we visualize the representative results in Figures 1–6, with AEN, EN, GL, and SCAD shown in red, blue, green, and yellow, respectively.

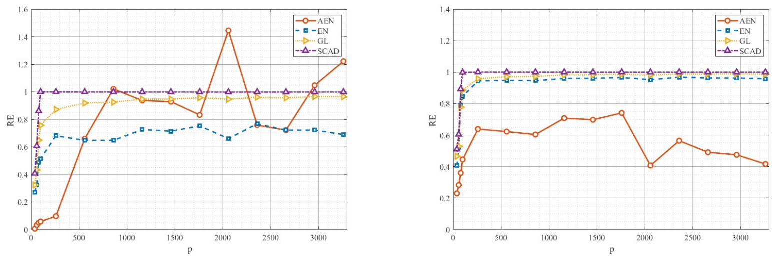

FIGURE 6. Comparison the RE of models at $\sigma = 0.5$ (left) and $\sigma = 2$ (right).

In Figures 1 to 3, we compare the estimated value β and the true value β^* of the four methods for different variances ($\sigma = 0.5, 1$ and 2) . It can be seen from Figures 1 to 3 that the estimated value of our proposed method is very close to the true value. Therefore, in high-dimensional data, the AEN method is superior to other methods.

In Figure 4, we compare the mean square error obtained by the four methods at different variances $\sigma = 0.5$ and 2. It can be seen that the mean square error of the four methods basically decreases with the increase of dimension, and the mean square error of AEN method remains at a low level, which is better than other methods.

In Figure 5, we compare the relative error between the estimated value β and the true value β^* obtained by four methods at $\sigma = 0.5$ and 2. It can be seen that the AEN method has stable performance and low RE in different dimensions. It is clear that AEN outperforms the other three methods in computational performance and relative error for high-dimensional problems.

In Figure 6, we compare the relative error between the estimated value of the model's objective function and the real value obtained by the four methods at $\sigma = 0.5$ and 2. When dealing with high-dimensional and high variance data, AEN has better computational performance and smaller relative error. It shows the advantages of AEN in dealing with high-dimensional problems. It can be seen that the EN, GL, and SCAD methods are more robust than the AEN method when dealing with low dimensional and low variance data.

In conclusion, numerical experiments show that the adaptive elastic net partially linear varying coefficient model proposed in this paper performs well in fitting high-dimensional data, and the fitting effect is better than other methods in high-dimensional data, so it has a certain applicability.

6. Application example

In order to further illustrate the effectiveness of the method proposed in this paper, we conducted an empirical analysis on the housing price data from Boston in 1970. The data can be accessed at boston_corrected.txt. This data set contains the median values of 506 owner occupied houses in the U.S. Census area. The variable settings are shown in Table 4. Due to the asymmetric distribution of LSTAT, we used its square root, denoted as $T_i = \sqrt{\text{LSTAT}}$.

We standardized all variables and established the model as follows:

$$Y_i = X_{1i}\beta_1 + X_{2i}\beta_2 + X_{3i}\beta_3 + Z_{1i}\alpha_1(T_i) + Z_{2i}\alpha_2(T_i) + Z_{3i}\alpha_3(T_i) + \varepsilon_i, \quad i = 1, 2, \ldots, 506. \tag{6.1}$$

In the experiment, we used 70% of the dataset as the training set and the remaining 30% as the test set. For the nonparametric part of model (6.1), we continued to use the local polynomial method for estimation. This experiment employed the median absolute error (MAE) and the standard error (SE) to measure the performance of the predictive model. MAE measures the deviation magnitude between predicted and actual values; it also

TABLE 4. Variable settings.

Variable	Definition
Y_i	Median value of owner-occupied homes (MEDV)
T_i	Percentage of lower status population (LSTAT)
X_1	Student-teacher ratio per town (PTRATIO)
X_2	Nitric oxide concentration (NOX)
X_3	Proportion of owner-occupied homes built before 1940 (AGE)
Z_1	Full value property tax rate per room (TAX)
Z_2	Average number of rooms per dwelling (RM)
Z_3	Per capita crime rate in the town (CRIM)

TABLE 5. MAE and SE for Boston housing price prediction.

Variable	MAE	SE
PTRATIO	0.16056	0.68623
NOX	0.19017	0.73302
AGE	0.18544	0.73576
TAX	0.28046	0.73257
RM	0.22849	0.84777
CRIM	0.29248	1.8997
All variables	0.20692	0.73137

reflects the volatility and uncertainty of predictions. The expressions for MAE and SE are as follows:

$$MAE = \text{median}\left(|y_1 - \hat{y}_1|, |y_2 - \hat{y}_2|, \ldots, |y_n - \hat{y}_n|\right),$$

$$SE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2},$$

where y_i is the predicted value, $\hat{y}_i$ is the true value, and n is the number of samples.

The MAE and SE results for the variables are shown in Table 5.

The results of Table 5 show that the median absolute error (MAE) is low, indicating high accuracy of the model in predicting Boston housing prices. Meanwhile, the standard error (SE) is also low, confirming the consistency and reliability of the predictions. In summary, the model proposed in this paper demonstrates significant effectiveness and robustness in housing price prediction.

7. CONCLUSION

This paper studies the partially linear varying coefficient model on high-dimensional settings. We propose an adaptive elastic net regularization model to estimate the parameters, then use the ADMM to solve the model, design the corresponding algorithm framework, further proving the convergence of the algorithm. Meanwhile, numerical experiments on high-dimensional data sets verify the effectiveness of the algorithm. It not only enriches the theoretical basis of partially linear varying coefficient model, but also provides a new perspective and method for high-dimensional data analysis.

Acknowledgments

The research was partially supported by the National Natural Science Foundation of China (Grant Nos. 12101195, 12071112), Provincial Demonstration Course Project of "Integration of Specialization and Innovation" and the Provincial first-class undergraduate curriculum project of mathematical model.

Data availability statement

The code used in this paper is available online in a Github repository: https://github.com/jiejie0326/code [30].

References

[1] T. Hastie and R. Tibshirani, Varying-coefficient models. *J. Roy. Statist. Soc. Ser. B: Statist. Methodol.* **55** (1993) 757–779.

[2] W. Zhang, S.Y. Lee and X. Song, Local polynomial fitting in semivarying coefficient model. *J. Multivariate Anal.* **82** (2002) 166–188.

[3] H. Hong, G. Ju, Q. Li, *et al.*, Varying-coefficient spatial dynamic panel data models with fixed effects: theory and application. *J. Econometrics* **245** (2024) 105883.

[4] Z. Qiu and Y. Zhou, Partially linear transformation models with varying coefficients for multivariate failure time data. *J. Multivariate Anal.* **142** (2015) 144–166.

[5] L. Li and T. Greene, Varying coefficients model with measurement error. *Biometrics* **64** (2008) 519–526.

[6] C. Wu, X. Shi, Y. Cui, *et al.*, A penalized robust semiparametric approach for gene–environment interactions. *Statist. Med.* **34** (2015) 4016–4030.

[7] J.Z. Huang, C.O. Wu and L. Zhou, Polynomial spline estimation and inference for varying coefficient models with longitudinal data. *Statistica Sinica* **14** (2004) 763–788.

[8] T. Qingguo, Robust estimation for spatial semiparametric varying coefficient partially linear regression. *Statist. Pap.* **56** (2015) 1137–1161.

[9] Z. Huang and R. Zhang, Empirical likelihood for nonparametric parts in semiparametric varying-coefficient partially linear models. *Statist. Probab. Lett.* **79** (2009) 1798–1808.

[10] A. Feng, X. Chang, Y. Shang, *et al.*, Application of the ADMM algorithm for a high-dimensional partially linear model. *Mathematics* **10** (2022) 4767.

[11] X. Cai, L. Xue and Z. Wang, Robust estimation with modified Huber's function for functional linear models. *Statistics* **54** (2020) 1276–1286.

[12] X. Cai, L. Xue and F. Lu, Robust estimation with a modified Huber's loss for partial functional linear models based on splines. *J. Korean Statist. Soc.* **49** (2020) 1214–1237.

[13] H. Sun and Q. Liu, Robust orthogonal empirical likelihood for partial linear models based on modified Huber's loss function. *J. Syst. Sci. Math. Sci.* **42** (2022) 1330.

[14] Q. Li, C.J. Huang, D. Li, *et al.*, Semiparametric smooth coefficient models. *J. Bus. Econ. Statist.* **20** (2002) 412–422.

[15] R. Tibshirani, Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B: Statist. Methodol.* **58** (1996) 267–288.

[16] H. Zou, The adaptive lasso and its oracle properties. *J. Am. Statist. Assoc.* **101** (2006) 1418–1429.

[17] K. Knight and W. Fu, Asymptotics for lasso-type estimators. *Ann. Statist.* **28** (2000) 1356–1378.

[18] R. Tibshirani, M. Saunders, S. Rosset, *et al.*, Sparsity and smoothness via the fused lasso. *J. Roy. Statist. Soc. Ser. B: Statist. Methodol.* **67** (2005) 91–108.

[19] J. Fan and R. Li, Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Statist. Assoc.* **96** (2001) 1348–1360.

[20] H. Zou and T. Hastie, Regularization and variable selection via the elastic net. *J. Roy. Statist. Soc. Ser. B: Statist. Methodol.* **67** (2005) 301–320.

[21] M. Zhao, A. Feng, J. Zhou, *et al.*, Optimization study of high-dimensional varying coefficient partially linear model based on elastic network. *Eng. Sci. Technol.* **55** (2024) 101731.

[22] M. Yuan and Y. Lin, Model selection and estimation in regression with grouped variables. *J. Roy. Statist. Soc. Ser. B: Statist. Methodol.* **68** (2006) 49–67.

[23] J. Friedman, T. Hastie and R. Tibshirani, A note on the group lasso and a sparse group lasso. arXiv preprint arXiv:1001.0736 (2010).

[24] Y. Zhou, R. Jin and S.C.H. Hoi, Exclusive lasso for multi-task feature selection, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (2010) 988–995.

[25] S. Petry, C. Flexeder and G. Tutz, Pairwise fused lasso. Technical Report (2011).

[26] Y. She, *Sparse Regression with Exact Clustering*. Ph.D. Thesis, Stanford University (2008).

[27] H. Zou and H.H. Zhang, On the adaptive elastic-net with a diverging number of parameters. *Ann. Statist.* **37** (2009) 1733–1751.

[28] B. He and X. Yuan, On non-ergodic convergence rate of Douglas–Rachford alternating direction method of multipliers. *Numer. Math.* **130** (2015) 567–577.

[29] H. Wang, G. Zou and A.T.K. Wan, Model averaging for varying-coefficient partially linear measurement error models. *Electron. J. Statist.* **6** (2012) 1017–1039.

[30] J. Zhou, *et al.* Matlab code for "A new class of high-Dimensional partially linear varying coefficient model and its applications" (2025). https://github.com/jiejie0326/code.