A Sparse Semismooth Newton-Based Manifold Algorithm for Partial Least Squares

1st Xianchao Xiu
School of Mechatronic Engineering and Automation
Shanghai University
Shanghai 200444, China
xcxiu@shu.edu.cn

2nd Ruijie Liu
School of Optical-Electrical and Computer Engineering
University of Shanghai for Science and Technology
Shanghai 200093, China
liurj@usst.edu.cn

3rd Zhonghua Miao
School of Mechatronic Engineering and Automation
Shanghai University
Shanghai 200444, China
zhmiao@shu.edu.cn

2022 China Automation Congress (CAC) | 978-1-6654-6533-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/CAC57257.2022.10055654

***Abstract*—Partial least squares (PLS) is a famous dimensionality reduction tool, which is widely used in signal processing. Although a lot of solvers exist in the literature, to the best of our knowledge, there are no fast optimization algorithms available for high-domensional PLS problems. To this end, we first construct a sparse and robust PLS model with good prediction performance and efficient feature selection. More importantly, we develop a semismooth Newton-based manifold proximal gradient algorithm by exploiting the sparse structure. Numerical comparisons validate its higher efficiency over state-of-the-art algorithms.**

***Index Terms*—Partial least squares (PLS), sparse optimization, manifold proximal gradient algorithm, semismooth Newton.**

I. Introduction

Partial least squares (PLS), dating back to Wold [1], provides an alternative method to ordinary least squares (OLS). Generally speaking, PLS is a dimensionality reduction technique that is designed for seeking a basic latent regression between two datasets. In comparison with OLS, PLS often has better prediction performance and performs more efficient for high-dimensional data analysis [2]. Without exaggeration, PLS has become one of the most important multivariate statistical analysis methods [3], and it has received widespread attention in the fields such as chemometrics [4], economics [5], and genetics [6].

Mathematically, for two observation datasets X and Y with n samples and p, q variables, PLS has a basic assumption that there exist matrix decompositions given by $X = TP^\top + E$ and $Y = TQ^\top + F$, where T is a latent part, P, Q are loading parts, and E, F are residual parts. Furthermore, suppose $T = XW$ with the orthogonal constraint $W^\top W = I_r$ (r is often omitted without ambiguity). Then the classical PLS minimizes

$$\begin{aligned} \min_{W} \quad & -\mathrm{Tr}(W^\top X^\top Y Y^\top X W) \\ \text{s.t.} \quad & W^\top W = I. \end{aligned} \tag{1}$$

Once W is determined, it derives $Y = XR + F$, of which $R = WQ^\top$ is the regression coefficient with $Q = Y^\top T(T^\top T)^{-1}$. That's where the word "partial" comes from.

For PLS-related problems, there are two popular algorithms. The first is a nonlinear iterative partial least squares (NIPALS) algorithm [1]. It randomly initializes a group of latent components and then conducts nonlinear iterations until a stopping condition is satisfied. The second is a statistically inspired modification of PLS algorithm, called SIMPLS [2]. It mainly generalizes the univariate PLS form and maintains the latent components orthogonal to each other. From an optimization perspective, both NIPALS and SIMPLS can only generate suboptimal solutions. From a computational perspective, PLS involves minimizing (1), which is associated with a reduced-rank Procrustes rotation problem. Even though it can be calculated by calculating singular value decompositions, plenty of time is required for high-dimensional dataset. During the past few decades, manifold optimization has been extensively researched and the performance is promising; see [7] for signal processing and [8] for image processing. Based on this, Chen *et al.* [9] developed an efficient algorithm to transform SIMPLS into an optimization problem over generalized Stiefel manifolds. It has been illustrated that manifold algorithms can substantially accelerate the optimization process. However, there is no convergence guarantee.

Although PLS enjoys good prediction performance, it cannot achieve variable selection, which means that some selected variables do not contribute to prediction, and even bring certain negative effects [10]. Inspired by advances in sparse representation, Chun and Keleş [6] considered a sparse PLS model by imposing the ℓ_1-norm induced sparsity in the dimensionality reduction step. The core idea is that the ℓ_1-norm can eliminate small residuals, and then improve the variable interpretability and reliability. After that, a number of spare PLS variants are proposed by adding different regularization terms [11]. However, the dataset we are dealing with usually has a certain structure, so it will be much better to capture its characteristics by properly exploiting and using this special structure. Therefore, an interesting question naturally arises: is it possible for us to derive a generalization of sparse PLS and develop a fast optimization algorithm.

This paper will give an affirmative answer. In optimization model, a novel sparse PLS form is constructed by integrating the $\ell_{2,1}$-norm, which is used to facilitate structured sparsity and then improve the ability of feature selection [12]. In numerical algorithm, by fully exploiting the $\ell_{2,1}$-norm induced sparsity, semismooth Newton techniques [13] can be used to accelerate the manifold optimization. Compared with [6], [9],

the targets of this paper are

1) To construct a sparse and robust PLS model.
2) To develop a fast manifold optimization algorithm.
3) To demonstrate the superiority on numerical examples.

II. Sparse PLS

A. Problem Formulation

Instead of applying the conventional ℓ_1-norm, we construct a novel sparse PLS formulation as

$$\begin{aligned} \min_{W} \ & -\mathrm{Tr}(W^\top X^\top Y Y^\top X W) + \lambda \|W\|_{2,1} \\ \text{s.t.} \ & W^\top W = I, \end{aligned}$$

where $\|\cdot\|_{2,1}$ is the $\ell_{2,1}$-norm (sum of ℓ_2-norm of all rows), and $\lambda > 0$ is the regularization parameter. Through choosing different λ, this model can drop out some zero-rows of W so as to exclude certain unimportant features of X.

Following similar arguments in SIMPLS, it is easy to obtain the regression coefficient R by revisiting the latent component $T = XW$. As mentioned in [9], to make the information in R (related to P and Q) more independent, one can enforce an orthogonal constraint as $T^\top T = W^\top X^\top X W = I$. Therefore, it derives a undeflated optimization problem given by

$$\begin{aligned} \min_{W} \ & -\mathrm{Tr}(W^\top X^\top Y Y^\top X W) + \lambda \|W\|_{2,1} \\ \text{s.t.} \ & W^\top X^\top X W = I. \end{aligned} \tag{2}$$

Although it has a similar structure with the existing sparse PLS variants in the literature, the model proposed in (2) differs in the following two ways:

- It introduces the $\ell_{2,1}$-norm to promote structured sparsity of W, which is much more efficient in feature selection and robust to outliers than the existing ℓ_1-norm in [6] and the vectored group sparsity in [14].
- It incorporates prior information of the given dataset X rather than directly applying $W^\top W = I$, thereby making the representation more discriminative.

Therefore, our proposed sparse PLS model can be seen as a PLS generalization. It is acknowledged that there exist some optimization algorithms, but all of them work slowly for large-scale sparse PLS problems. Below, we will discuss how to develop a fast optimization algorithm for solving problem (2), and further investigate how to apply a semismooth Newton technique for the resulting subproblems in Section III.

B. Optimization Framework

To begin with this subsection, we first suppose that $F(W) = -\mathrm{Tr}(W^\top X^\top Y Y^\top X W)$ and $\mathcal{M} = \{W \mid W^\top X^\top X W = I\}$. Then, problem (2) can be rewritten as

$$\min_{W \in \mathcal{M}} \ F(W) + \lambda \|W\|_{2,1}. \tag{3}$$

Notice that the objective has a composite structure, including a smoothing term $F(W)$ and a nonsmooth term $\|W\|_{2,1}$, along with a manifold constraint $W \in \mathcal{M}$. Therefore, problem (3) is both nonconvex and nonsmooth, which makes it difficult to be optimized directly. With the help of proximal gradient methods, one can seek an approximate solution by considering the following optimization problem

$$\begin{aligned} \min_{W \in \mathcal{M}} \ & F(W_k) + \langle \nabla F(W_k), W - W_k \rangle \\ & + \frac{1}{2t}\|W - W_k\|_F^2 + \lambda \|W\|_{2,1}. \end{aligned} \tag{4}$$

Here, $\nabla F(W_k)$ is the gradient of F at pint W_k, $t > 0$ is the weighting parameter with an upper bound $t \le 1/L$, of which L is the Lipschitz constant of ∇F. In fact, it is a linearization technique whose key idea is to consider an easier version (4) instead of solving (3). Denote the tangent of $\mathcal{M}$ be $\mathrm{T}_{W_k}\mathcal{M}$. It is possible to find the descent direction D_k in the kth iteration by solving

$$\begin{aligned} \min_{D} \ & \langle \nabla F(W_k), D \rangle + \frac{1}{2t}\|D\|_F^2 + \lambda \|W_k + D\|_{2,1} \\ \text{s.t.} \ & D \in \mathrm{T}_{W_k}\mathcal{M}. \end{aligned} \tag{5}$$

Overall, the optimization framework for solving problem (3) (equivalent to problem (2)) can be given in Algorithm 1. Here, Step 2 is adopted to ensure D lying on the manifold $\mathcal{M}$ for an arbitrary stepsize $\alpha_k > 0$, where the retraction is performed to bring it back to $\mathcal{M}$.

Algorithm 1 Optimization Framework for (3)

Input: Datasets X, Y, parameters λ, t, γ
Initialization: W_0
While not converged do

1: Solve problem (5)
2: Set $\alpha = 1$, update

$$W_{k+1} = \mathcal{R}_{W_k}(\alpha_k D_k)$$

where $\alpha_k = \gamma^k \alpha$ and k is smallest nonnegative integer such that

$$F(W_{k+1}) \le F(W_k) - \frac{\alpha_k}{2t}\|D_k\|_F^2$$

3: Check convergence

End while

To end this section, three lemmas are introduced, which are necessary for the following analysis.

Lemma 2.1. *For the minimization problem*

$$\min_{W} \ \frac{1}{2}\|W - T\|_F^2 + \lambda \|W\|_{2,1},$$

it has a unique optimal solution characterized by the proximal mapping Prox$_{\lambda\|\cdot\|_{2,1}}(\cdot)$, which is given by

$$\mathrm{Prox}_{\lambda\|\cdot\|_{2,1}}(T_i) = \frac{T_i}{\|T_i\|_2} \max\{0, \|T_i\|_2 - \lambda\}$$

with T_i representing the ith row of T; see [15].

Lemma 2.2. *For three matrices $A \in \mathbb{R}^{n\times p}$, $B \in \mathbb{R}^{p\times q}$, and $C \in \mathbb{R}^{q\times r}$, it holds*

$$\mathrm{vec}(ABC^\top) = (C \otimes A)\mathrm{vec}(B),$$

with vec$(\cdot)$ denoting the vectorization of a matrix and $\otimes$ denotes the Kronecker product.

Lemma 2.3. *For a matrix $A \in \mathbb{R}^{n\times p}$, there exists a commutation matrix $K_{np} \in \mathbb{R}^{np\times np}$ satisfying*

$$K_{np}\text{vec}(A) = \text{vec}(A^\top).$$

Moreover, for another matrix $B \in \mathbb{R}^{m\times q}$, it further has the following property

$$K_{mn}(A \otimes B)K_{pq} = B \otimes A.$$

III. Semismooth Newton Method

A. Solution for Subproblem (5)

Obviously, the main computational load is solving subproblem (5) in Step 1. Now, it comes to compute a proximal gradient step restricted on the tangent space $\text{T}_{W_k}\mathcal{M}$. According to the definitions of generalized Stiefel manifolds and tangent spaces, the explicit formulation of $D \in \text{T}_{W_k}\mathcal{M}$ can be characterized as

$$\text{T}_{W_k}\mathcal{M} = \{D \mid D^\top X^\top XW_k + W_k^\top X^\top XD = 0\}.$$

Thus, subproblem (5) can be transformed into the form of

$$\begin{aligned} \min_D \ & \langle \nabla F(W_k), D\rangle + \frac{1}{2t}\|D\|_F^2 + \lambda\|W_k + D\|_{2,1} \\ \text{s.t.} \ & D^\top X^\top XW_k + W_k^\top X^\top XD = 0, \end{aligned} \tag{6}$$

and the Lagrangian is given by

$$\begin{aligned} \mathcal{L}(D,\Lambda) = \langle \nabla F(W_k), D\rangle + \frac{1}{2t}\|D\|_F^2 + \lambda\|W_k + D\|_{2,1} \\ - \langle \Lambda, D^\top X^\top XW_k + W_k^\top X^\top XD\rangle, \end{aligned}$$

where Λ is a Lagrange multiplier. Recall the first-order optimal condition, it derives the following Karush-Kuhn-Tucker system of problem (6) as

$$\begin{cases} 0 \in \partial_D \mathcal{L}(D,\Lambda), & \text{(7a)} \\ 0 = D^\top X^\top XW_k + W_k^\top X^\top XD, & \text{(7b)} \end{cases}$$

where $\partial_D\mathcal{L}(D,\Lambda)$ is gradient of $\mathcal{L}(D,\Lambda)$ in terms of D. From Lemma 2.1, the first condition (7a) can be expanded into

$$D(\Lambda) = \text{Prox}_{\lambda t\|\cdot\|_{2,1}}(H(\Lambda)) - W_k \tag{8}$$

with $H(\Lambda) = W_k - t(\nabla F(W_k) - 2X^\top XW_k\Lambda)$. By substituting (8) into the second condition (7b), it implies that

$$D(\Lambda)^\top X^\top XW_k + W_k^\top X^\top XD(\Lambda) = 0. \tag{9}$$

Since the semismooth Newton method has demonstrated to be highly efficient in sparse representation, our goal is to apply it for solving problem (9) and accelerate the proposed Algorithm 1. Denote $G(\Lambda) = D(\Lambda)^\top X^\top XW_k + W_k^\top X^\top XD(\Lambda)$. Thus the vectorization form of $G(\Lambda)$ is characterized by

$$\begin{aligned} \text{vec}(G(\Lambda)) &= ((X^\top XW_k)^\top \otimes I_r)\text{vec}(D(\Lambda)^\top) \\ &\quad + (I_r \otimes (X^\top XW_k)^\top)K_{rp}\text{vec}(D(\Lambda)^\top) \\ &= (I_{r^2} + K_{rr})((X^\top XW_k)^\top \otimes I_r)\text{vec}(D(\Lambda)^\top), \end{aligned}$$

where these equivalent transformations come from Lemma 2.2 and Lemma 2.3. Furthermore, it is achieved that

$$\begin{aligned} \text{vec}(G(\text{vec}(\Lambda))) = (I_{r^2} + K_{rr})((X^\top XW_k)^\top \otimes I_r) \\ \big(\text{Prox}_{\lambda t\|\cdot\|_{2,1}}(\text{vec}(W_k - t\nabla F(W_k)) \\ + 2t((X^\top XW_k) \otimes I_r)\text{vec}(\Lambda)\big) - \text{vec}(W_k^\top)\big). \end{aligned}$$

Afterwards, it is necessary to study the generalized Jacobian of $\text{Prox}_{\lambda t\|\cdot\|_{2,1}}(H(\text{vec}(\Lambda)))$, that is, $\text{Prox}_{\lambda t\|\cdot\|_{2,1}}(\text{vec}(W_k - t\nabla F(W_k)) + 2t((X^\top XW_k)\otimes I_r)\text{vec}(\Lambda)))$. Assume that $U = \partial\text{Prox}_{\lambda t\|\cdot\|_{2,1}}(H(\text{vec}(\Lambda))$. Invoking the properties of proximal mappings, it should have $U = 2\text{Diag}(U_1,\ldots,U_p)$, of which U_i $(i = 1,\ldots,p)$ is defined as

$$U_i = \begin{cases} I_r - \frac{\lambda t}{\|h_j\|_2}(I_r - \frac{h_j h_j^\top}{\|h_j\|_2^2}), & \text{if } \|h_j\|_2 \geq \lambda t, \\ 0, & \text{otherwise,} \end{cases}$$

where h_j is the jth column of $H(\Lambda)^\top$. Hence, the Jacobian of $\text{vec}(G(\text{vec}(\Lambda)))$ can be described as

$$\begin{aligned} J(\text{vec}(\Lambda)) &= \partial\text{vec}(G(\text{vec}(\Lambda))) \\ &= (I_{r^2} + K_{rr})((X^\top XW_k)^\top \otimes I_r) \\ &\quad U((X^\top XW_k) \otimes I_r) \\ &= 2(I_{r^2} + K_{rr})((X^\top XW_k)^\top \otimes I_r) \\ &\quad \begin{bmatrix} U_1 & & \\ & \ddots & \\ & & U_p \end{bmatrix} ((X^\top XW_k) \otimes I_r). \end{aligned}$$

Therefore, the Newton's search direction d_k can be approximated obtained by

$$(J(\text{vec}(\Lambda_k) + \epsilon I)d = -\text{vec}(G(\text{vec}(\Lambda_k))), \tag{10}$$

where $\epsilon > 0$ is the regularization parameter to guarantee its invertible. Moreover, Λ_k can be calculated by

$$\text{vec}(\Lambda_{k+1}) = \text{vec}(\Lambda_k) + d_k. \tag{11}$$

This, together with the aforementioned relation (8), derives the semismooth Newton method for subproblem (5) in Algorithm 2, where the convergence in Step 3 is defined as $\|G(\Lambda)\|_F \leq 10^{-5}$. We would like to point out that the main calculation cost in Step 1 is $\mathcal{O}(r^6)$ for each iteration. Compared with [9], it is much cheaper because r is much smaller than p.

Algorithm 2 Semismooth Newton method for (5)

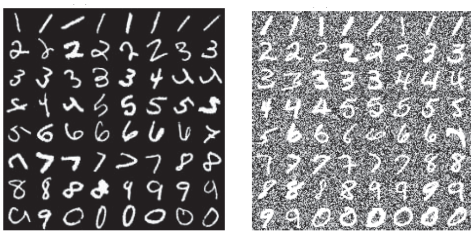
Input: Datasets X, Y, parameters λ, t, ϵ.
Initialization: Λ_0
While not converged do
1: Search the direction by (10)
2: Update Λ_{k+1} by (11)
3: Check convergence
End while
Compute (8)
Output: D

TABLE I
COMPARISON OF ALGORITHMS FOR PROBLEMS WITH $n \leq p$.

Dimensions	SIMPLS	SPLSRGSt	This work
(100;100;100)	**0.01**	0.05	0.05
(100;500;500)	0.23	0.09	**0.06**
(100;1,000;1,000)	1.14	0.32	**0.14**
(100;5,000;5,000)	295.19	7.31	**2.68**
(100;10,000;10,000)	1456.88	40.92	**6.45**

TABLE II
COMPARISON OF ALGORITHMS FOR PROBLEMS WITH $n \geq p$.

Dimensions	SIMPLS	SPLSRGSt	This work
(1,000;1,000;1,000)	10.08	0.17	**0.08**
(5,000;1,000;1,000)	11.24	0.28	**0.15**
(10,000;1,000;1,000)	15.38	0.61	**0.24**
(50,000;1,000;1,000)	16.95	0.84	**0.56**
(100,000;1,000;1,000)	19.73	1.05	**0.79**



(a) (b)

Fig. 1. Images of the MNIST dataset: (a) view 1, (b) view 2.

TABLE III
CLUSTERING RESULTS FOR THE MNIST DATASET.

	SIMPLS	SPLSRGSt	This work
ACC	68.32%	73.21%	**76.14%**
ERR	31.65%	26.79%	**25.86%**

B. Convergence Analysis

In order to characterize the convergence, we first given the definition of ε-stationary point.

Definition 3.1. *In this paper, We call W is a stationary point of problem* (3) *((equivalent to problem* (2)*) if D returned by subproblem* (5) *satisfies $\|D_k\|_F^2 \leq \varepsilon$.*

Next, it is possible to establish the convergence property for the proposed Algorithm 1.

Theorem 3.2. *Suppose that $\{W_k\}$ is a generated sequence of the proposed Algorithm 1 and the Step 1 is solved by Algorithm 2. Then the sequence converges to a stationary point of problem* (2) *after finite iterations.*

IV. NUMERICAL VERIFICATION

A. Simulation Examples

To verify the efficiency of our proposed algorithm, this section provides simulation experiments in different dimensions. First, two datasets $X,\ Y$ are constructed as

$$X = \mathbf{t}(\mathbf{p}+\mathbf{e})^\top,\ Y = \mathbf{t}(\mathbf{q}+\mathbf{f})^\top,$$

where $\mathbf{t} \in \mathbb{R}^{100i}$ and $\mathbf{e},\ \mathbf{f} \in \mathbb{R}^{100j}$ are the random vectors generated from standard normal distributions, $\mathbf{p},\ \mathbf{q} \in \mathbb{R}^{100j}$ are the special vectors given by

$$\mathbf{p} = [\underbrace{1\cdots1}_{10j}\ \underbrace{-1\cdots-1}_{10j}\ \underbrace{0\cdots0}_{80j}]^\top,$$
$$\mathbf{q} = [\underbrace{0\cdots0}_{80j}\ \underbrace{1\cdots1}_{10j}\ \underbrace{-1\cdots-1}_{10j}]^\top,$$

where $i,\ j$ are the positive integers. In addition, the reduced dimension of W is $r = 10j$.

For SIMPLS [6] and SPLSRGSt [9], codes are downloaded from the authors' homepage and set to default values. To be fair, all three algorithms choose the same initial point, i.e., the solution of PLS, and stop until the relative difference is less than 10^{-5} or the maximum iteration is 10^3. Table I and Table II report comparisons for $n \leq p$ and $n \geq p$, respectively. Here, the first column lists the problem dimension $(n; p; q) = (100i; 100j; 100j)$, and the others list the CPU time in seconds. In this study, all results are averaged over 20 replications under $\lambda = 1$ for all testing cases. It can be seen that compared with SIMPLS, the manifold optimization algorithms, i.e., SPLSRGSt and this work, are much faster. This is because SIMPLS needs to calculate the singular value decomposition in each iteration. As the problem dimensions increase, the advantages of our proposed algorithm become more pronounced. Especially for the case of $(100; 10,000; 10,000)$, the efficiency is nearly 200 times better than SIMPLS and 6 times better than SPLSRGSt. This shows that the semismooth Newton technique is able to exploit the structured sparsity and is more powerful for large-scale sparse PLS problems.

B. Image Clustering

To verify the performance of feature selection, this section conducts a practical application of image clustering on the MNIST data set [16]. It has 10 classes of 28×28 grayscale handwritten digits with 70,000 images; see Fig. 1(a) for some images. The data set is first scaled and rotated, then added Gaussian noise; see Fig. 1(b) for the corresponding images. Finally, the data set is cut into a training set containing 50,000 images, a tuning set containing 10,000 images, and a testing set containing 10,000 images.

In order to estimate the cluster performance, two common measurements are adopted, namely clustering accuracy (ACC) and classification error (ERR). In this study, λ is chosen carefully by the five-fold cross-validation technique. Table III gives the clustering results achieved by all compared algorithms. Obviously, the proposed algorithm achieves the best results. To be specific, compared with SIMPLS, the increase of ACC is 7.72% and the reduction of ERR is 5.79%. Furthermore, Fig. 2 shows the clustering performance through t-SNE. Although all of them fail to complete this task, the proposed algorithm outperforms SIMPLS and SPLSRGSt, which confirms the promising performance of this work.

V. CONCLUSION

In this paper, we have proposed an $\ell_{2,1}$-norm regularized sparse PLS model, which, to the best knowledge, has not been

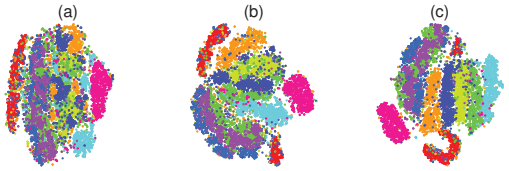


Fig. 2. Clustering performance: (a) SIMPLS, (b) SPLSRGSt, (c) This work.

investigated yet. Furthermore, we have developed an efficient semismooth Newton-based manifold proximal gradient algorithm to alleviate computational burdens significantly, thereby accelerating the optimization procedure. The computational efficiency and feature selection ability of the proposed algorithm were verified by sufficient experiments.

Naturally, it is interesting to extend this work to deep learning-based frameworks and developing efficient optimization algorithms.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grand 12001019.

REFERENCES

[1] H. Wold, "Estimation of principal components and related models by iterative least squares," *Multivariate Analysis*, pp. 391–420, 1966.
[2] S. De Jong, "SIMPLS: An alternative approach to partial least squares regression," *Chemometrics and Intelligent Laboratory Systems*, vol. 18, no. 3, pp. 251–263, 1993.
[3] V. Vinzi, W. Chin, J. Henseler, H. Wang *et al.*, *Handbook of Partial Least Squares*. Springer, 2010, vol. 201.
[4] G. Li, S. Qin, and D. Zhou, "Geometric properties of partial least squares for process monitoring," *Automatica*, vol. 46, no. 1, pp. 204–210, 2010.
[5] Z. Wang, B. Zhang, B. Wang *et al.*, "The moderating role of corruption between economic growth and CO2 emissions: Evidence from brics economies," *Energy*, vol. 148, pp. 506–513, 2018.
[6] H. Chun and S. Keleş, "Sparse partial least squares regression for simultaneous dimension reduction and variable selection," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 1, pp. 3–25, 2010.
[7] M. Huang, S. Ma, and L. Lai, "Robust low-rank matrix completion via an alternating manifold proximal gradient continuation method," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2639–2652, 2021.
[8] L. Xu, X. Zeng, B. Zheng, and W. Li, "Multi-manifold deep discriminative cross-modal hashing for medical image retrieval," *IEEE Transactions on Image Processing*, vol. 31, pp. 3371–3385, 2022.
[9] H. Chen, Y. Sun, J. Gao, Y. Hu, and B. Yin, "Solving partial least squares regression via manifold optimization approaches," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 2, pp. 588–600, 2019.
[10] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3133–3181, 2014.
[11] Q. Jiang, X. Yan, H. Yi, and F. Gao, "Data-driven batch-end quality modeling and monitoring based on optimized sparse partial least squares," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 5, pp. 4098–4107, 2020.
[12] F. Nie, H. Huang, X. Cai, and C. Ding, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization," *Advances in Neural Information Processing Systems*, vol. 23, 2010.
[13] Y. Liu, Z. Wen, and W. Yin, "A multiscale semi-smooth Newton method for optimal transport," *Journal of Scientific Computing*, vol. 91, no. 2, pp. 1–29, 2022.
[14] B. Liquet, P. De Micheaux, B. Hejblum, and R. Thiébaut, "Group and sparse group partial least square approaches applied in genomics context," *Bioinformatics*, vol. 32, no. 1, pp. 35–42, 2016.
[15] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.
[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.