Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 16, No. 2, 2026

Improved Sparse Principal Component Analysis Algorithm based on Non-convex Optimization

Weihao Li

Xi'an Innovation College of Yan'an University, Xi'an, Shaanxi, 710100, China

Abstract: In order to reduce the problems of poor robustness and high computational complexity in Convex Sparse Principal Component Analysis (CSPCA) when dealing with high-dimensional data, we propose a new improved convex sparse principal component analysis algorithm, that is, a non-convex SPCA algorithm based on the $l_{2,p}$-norm form: $l_{2,p}$-NSPCA. The proposed $l_{2,p}$-NSPCA is a two-stage algorithm. In the first stage, performing CSPCA without a low-rank penalty term and introducing a generalized inverse lemma on the optimization objective model, to achieve the goal of improving computing efficiency; In the second stage, matrix factorization is performed first, and then the $l_{2,p}$-norm is adopted as a new low-rank non-convex regularization term, so as to better improve the robust performance of the algorithm. We use $l_{2,p}$-NSPCA for unsupervised feature selection, and the comparative experimental results on five gene expression data show that the algorithm we propose is less affected by parameters, has better feature selection performance, and runs faster.

Keywords: Non-convex Optimization; Convex Sparse Principal Component Analysis; Generalized Inverse Lemma of Matrices; Unsupervised Feature Selection.

1. Introduction

In applications such as computer vision, data mining, pattern recognition, and machine learning, such as face recognition and genetic data analysis, the input data set is located in an observation space of thousands of dimensions, and the very high data dimensionality limits many practical applications, direct analysis of high-dimensional data is not only computationally expensive, but also difficult to process [1]. At the same time, with the increase of data dimensionality, the noise data and redundant information in the original high-dimensional data may increase significantly, leading to deviations in the results of data analysis, which brings great challenges to high-dimensional data processing. Therefore, how to process high-dimensional data more efficiently has become an urgent problem to be solved. A large number of studies have shown that dimensionality reduction of data is one of the important ways to analyze and process high-dimensional data. In the 1980s, Svante first proposed Principal Component Analysis (PCA), and used it for data dimensionality reduction[2]. PCA is a very popular unsupervised data processing and dimensionality reduction method. Its main idea is to map the dimensional data features to the k dimensional $(n \ll k)$, seek the linear combination of the original high-dimensional data features, and obtain an effective low-dimensional representation of high-dimensional data. However, because the new features of the data obtained by PCA are the linear combination of the original features of the data, they often lack interpretability. Subsequently, Zou et al[3]. proposed the sparse principal component analysis algorithm (Sparse Principal Component Analysis, SPCA), which expressed PCA as a regression-type optimization problem, and introduced sparse regularization terms, thus turning PCA into a feature Method of choosing. SPCA can not only be used for routine data analysis, but also can be effectively applied to gene expression array analysis. However, the SPCA algorithm is non-convex, and it is difficult to obtain the global optimal solution. When the local optimal solution is not the global optimal solution, the performance may change significantly. In view of this, Chang et al[4]. proposed Convex Sparse Principal Component Analysis (CSPCA). The CSPCA algorithm obtains a new SPCA algorithm by introducing a low-rank penalty item of the convex approximation of the nuclear norm in SPCA, and replacing the Frobenius norm (F-norm) in the SPCA loss function with the - norm. CSPCA is a globally optimal algorithm. The experimental results on a large number of data sets show that CSPCA has excellent feature selection performance and robustness to noise. However, there are two problems with CSPCA: the first is that the low-rank regularization term of the nuclear norm is introduced in the algorithm model, and the nuclear norm takes equal punishment on all singular values of the matrix, while the singular values of the matrix are actually The above represents the information in the matrix, which will cause some large singular values (important information in the data) to be reduced or even missing, resulting in approximation errors, which are usually not ideal in practical applications. The second is that the algorithm solution involves matrix inversion operations. When the data dimension is generally high, the computational complexity is high and the running time is long, which will limit the application range of CSPCA.

Aiming at the above two problems of CSPCA, this chapter proposes an algorithm based on non-convex optimization $l_{2,p}$ - NCSPCA (Non-convex CSPCA), that is $l_{2,p}$ - NSPCA algorithm. The $l_{2,p}$ -NSPCA algorithm is used in unsupervised feature selection, which can be divided into two stages. Firstly, in the first stage, the CSPCA algorithm without low-rank penalty term is used to perform initial dimensionality reduction processing on the data. In this stage, this chapter will use the generalized inverse lemma of the matrix to reduce the complexity of the algorithm, so as to improve the operational efficiency of the algorithm. In the second stage, the CSPCA algorithm with a low-rank penalty

item is executed to reduce the dimensionality of the data again. In the second stage, we first perform the matrix decomposition operation, and secondly, for the matrix low-rank regularization item in this stage, we use $l_{2,p}$ -norm $(2 < p < +\infty)$ replaces the original nuclear norm regularization term, making it a low-rank approximate penalty term of non-convex relaxation, so that it is comparable to the original nuclear norm convex relaxation while reducing the rank Than, the robust performance is superior, and the effect is superior. The comparative results of the experiment on five gene expression data show that the algorithm we proposed is less affected by parameters, and it shows competitiveness in the comparison of operating efficiency and clustering accuracy in unsupervised feature selection.

The rest of this article is organized as follows: Section 2 introduces the notations and definitions used in this paper, and briefly reviews three related works. Contains principal component analysis, sparse principal component analysis and convex-sparse principal component analysis algorithms. Section 3 introduces the $l_{2,p}$ - NCSPCA algorithm. Section 4 verifies the performance of $l_{2,p}$ - NCSPCA through experiments. Finally, we conclude in Section 5.

2. Related Work

2.1. Notations and Definitions

In recent years, matrix norms and matrix inner products have played the most basic role in data dimensionality reduction related algorithms, so this section mainly introduces several related and common matrix norms and matrix inner products. In this paper, all matrices are denoted by capital bold letters, and all vectors are denoted by lower case bold letters.

Definition 1: Let the matrices $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{m \times n}$, $\boldsymbol{a}_j$ and $\boldsymbol{b}_j$ represent the column vectors of the matrix $\boldsymbol{A}$ and $\boldsymbol{B}$ the matrix respectively, then the inner product of the matrix is defined as[5]:

$$\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \sum_{i=1}^{n} \langle \boldsymbol{a}, \boldsymbol{b} \rangle = Tr\left(\boldsymbol{A}^T \boldsymbol{B}\right). \tag{1}$$

And $Tr(\cdot)$ in formula (1) is called the trace function of the matrix.

Assuming that the matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$, represents $\sigma_i(\boldsymbol{X})$ the i th singular value of the matrix $\boldsymbol{X}$, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ represents the i th row and the j th column of the matrix $\boldsymbol{X}$ respectively, due to the existence of the inner product of the matrix $\boldsymbol{X}$, the nuclear norm of the matrix $\boldsymbol{X}$ [5], the Frobenius norm (F-norm), the $l_{2,1}$ -norm and $l_{2,p}$ -norm[5] of the matrix $\boldsymbol{X}$ are defined as:

$$\| \boldsymbol{X} \|_* = Tr\left(\left(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\right)^{\frac{1}{2}}\right) = \sum_{i=1}^{m} \sigma_i(\boldsymbol{X}), \tag{2}$$

$$\| \boldsymbol{X} \|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} | \boldsymbol{x}_{ij} |^2 \right)^{\frac{1}{2}}. \tag{3}$$

In addition, in the research related to feature selection, since the $l_{2,1}$ -norm and $l_{2,p}$ -norm of the matrix can improve the robustness of the algorithm, the $l_{2,1}$ -norm and $l_{2,p}$ -norm of the matrix are used in the objective function, and their definitions are respectively As follows:

$$\| \boldsymbol{X} \|_{2,1} = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} | \boldsymbol{x}_{ij} |^2 \right)^{\frac{1}{2}} = \sum_{i=1}^{m} \| \boldsymbol{x}^i \|_2, \tag{4}$$

$$\| \boldsymbol{X} \|_{2,p} = \left(\sum_{i=1}^{m} \left(\sum_{j=1}^{n} | \boldsymbol{x}_{ij} |^2 \right)^{\frac{p}{2}} \right)^{\frac{1}{p}} = \left(\sum_{i=1}^{m} \| \boldsymbol{x}_i \|_2^p \right)^{\frac{1}{p}}. \tag{5}$$

2.2. Rrincipal Component Analysis

PCA is a statistical method for data dimension reduction[6], which aims to seek the linear combination of the original high-dimensional data variables, so as to obtain the low-dimensional representation of high-dimensional data. PCA can be described as a regression-type optimization model, namely:

$$\min_{r(\boldsymbol{W})=k} \left\| \left(\boldsymbol{W}^T \boldsymbol{X} - \boldsymbol{X}\right)^T \right\|_F^2. \tag{6}$$

In formula (6), r is the rank of the matrix $\boldsymbol{W}$, that is, and a means the rank number of matrix w is d. PCA is solved by the method of least squares, which is extremely sensitive to noise. When the data contains noise, the PCA projection direction deviates from the desired optimal solution. In addition, while PCA reduces the dimensionality of data, the features may change, therefore, it cannot be used for feature selection.

2.3. Sparse Principal Component Analysis

The $l_{2,1}$ -norm of a matrix is shown to be able to make groups of matrices sparsify[6]. Therefore, SPCA can be described as the following optimization model:

$$\min_{r(\boldsymbol{W})=k} \left\| \left(\boldsymbol{W}^T \boldsymbol{X} - \boldsymbol{X}\right)^T \right\|_2^2 + \alpha \| \boldsymbol{W} \|_{2,1}. \tag{7}$$

Here, a is a non-negative regularization parameter, and the objective function is convex, The least squares loss function $\min_{r(\boldsymbol{W})=k} \left\| \left(\boldsymbol{W}^T \boldsymbol{X} - \boldsymbol{X}\right)^T \right\|_2^2$ is particularly sensitive to outliers, and the solution is a local optimal solution, which does not satisfy the global optimality.

2.4. Convex Sparse Principal Component Analysis

The $l_{2,1}$ -norm is proven to handle noisy data better [7], CSPCA changes the loss function in SPCA to $\min_{r(\boldsymbol{W})=k} \left\| \left(\boldsymbol{W}^T \boldsymbol{X} - \boldsymbol{X}\right)^T \right\|_{2,1}$, At the same time, the nuclear norm of $\boldsymbol{W}$ is added to limit $\boldsymbol{W}$ to a low-rank matrix. The algorithm can be described as the following optimization model:

$$\min_{\boldsymbol{W}} \| \left(\boldsymbol{W}^T \boldsymbol{X} - \boldsymbol{X}\right)^T \|_{2,1} + \alpha \| \boldsymbol{W} \|_{2,1} + \beta \| \boldsymbol{W} \|_*, \tag{8}$$

In the formula, β is the regularization parameter of $\boldsymbol{W}$ nuclear norm.

The $l_{2,1}$ -norm-based loss function $\min_{r(\boldsymbol{W})=k} \left\| \left(\boldsymbol{W}^T \boldsymbol{X} - \boldsymbol{X}\right)^T \right\|_{2,1}$ and regularization term $\| \boldsymbol{W} \|_{2,1}$ ensure the robustness to outliers and make $\boldsymbol{W}$ rows sparse, and the nuclear norm

$\|\boldsymbol{W}\|_*$ guarantees the low rank of $\boldsymbol{W}$. Because CSPCA is a convex optimization problem, the unique global optimal solution of CSPCA can be obtained by using the derivative method, and it shows good feature selection performance when used for unsupervised feature selection. However, CSPCA optimization involves matrix inversion process, and the main computational complexity is $O(d^3)$. Therefore, for high-dimensional dimensions, CSPCA has the problem of high computational complexity. The following will improve the SPCA algorithm to improve its computational efficiency while ensuring the performance of the algorithm.

3. $l_{2,p}$-NSPCA

3.1. Algorithm Principle of $l_{2,p}$-NSPCA

In view of the fact that the reason for the high computational complexity of CSPCA is mainly the optimization calculation of the penalty term of the nuclear norm and the theoretical deviation of the convex approximation of the nuclear norm[7], the proposed $l_{2,p}$-NSPCA algorithm will be divided into two stages: the first stage uses a robust SPCA performs unsupervised feature selection on the data to reduce the dimensionality of the data, and uses the generalized inverse lemma of the matrix to reduce the computational complexity; in the second stage, the matrix is first decomposed, and then the $l_{2,p}$ - norm $(2<p<+\infty)$ is used to replace the original nuclear norm regularization The transformation term makes it a non-convex relaxed low-rank approximation term, and finally performs feature selection on the dimensionality reduction data, so as to finally realize the feature selection on the original data.

Let $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n]\in \mathrm{R}^{d\times n}$ be the original data matrix, $\boldsymbol{x}^i\in \mathrm{R}^n\left(1\le i\le d\right)$ is the ith row of data, d is the number of rows, n is the total number of samples, the matrix $\boldsymbol{Y}\in \mathrm{R}^{d\times n}$ is the data after feature selection in the first stage of the original data matrix $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n]\in \mathrm{R}^{d\times n}$, the matrix $\boldsymbol{Z}\in \mathrm{R}^{d\times n}$ is the $l_{2,p}$-NSPCA algorithm in the The data after feature selection in the second stage. The two-stage description of the $l_{2,p}$-NSPCA algorithm is shown below.

(a) The first stage

The first stage of the $l_{2,p}$-NSPCA algorithm can be described as the following minimization problem:

$$\min_{r(\boldsymbol{W}')=k} \left\|\left(\boldsymbol{W}'^T\boldsymbol{X}-\boldsymbol{X}\right)^T\right\|_{2,1}+\lambda\|\boldsymbol{W}'\|_{2,1}, \tag{9}$$

Among them, $\boldsymbol{W}'\in \mathrm{R}^{d\times d}$ is the weight matrix of the first stage; $\boldsymbol{w}'^i$ represents the ith row of $\boldsymbol{W}'$, and λ is the parameter of $\|\boldsymbol{W}'\|_{2,1}$.

(b) The second stage

For the objective function of the second stage of the $l_{2,p}$-NSPCA algorithm, firstly, the dimensionality reduction data $\boldsymbol{Y}$ obtained in the first stage are substituted into the original CSPCA formula (8). Secondly, decompose the weight matrix $\boldsymbol{W}$ of the low-rank item first, so that $\boldsymbol{W}=\boldsymbol{U}\boldsymbol{V}^T$, $\boldsymbol{U}\in \mathrm{R}^{d\times r}$ and $\boldsymbol{V}\in \mathrm{R}^{d\times r}$ are the matrix factors after matrix $\boldsymbol{W}$ decomposition, and r are the inner dimensions of the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$. Secondly, the $l_{2,p}$-norm is used to replace the low-rank kernel norm item in the original CSPCA formula (8), then the objective function of the second stage of the $l_{2,p}$-NSPCA algorithm can be expressed as the following minimization problem:

$$\begin{aligned}&\min_{r(\boldsymbol{W})=k,\boldsymbol{U}\in \mathrm{R}^{d\times r},\boldsymbol{V}\in \mathrm{R}^{d\times r}} \left\|\left(\boldsymbol{W}^T\boldsymbol{Y}-\boldsymbol{Y}\right)^T\right\|_{2,1}+\alpha\|\boldsymbol{W}\|_{2,1}+\beta(\|\boldsymbol{U}\|_{2,p}^p+\|\boldsymbol{V}\|_{2,p}^p)\\ &s.t.\ \boldsymbol{W}=\boldsymbol{U}\boldsymbol{V}^T,\end{aligned} \tag{10}$$

where α and β are non-negative regularization parameters, and k represents the rank of the weight matrix $\boldsymbol{W}$.

3.2. Algorithm Solution of $l_{2,p}$-NSPCA

After the $l_{2,p}$-NSPCA algorithm is divided into two stages, the objective function formula (9) of the first stage is convex, so use the formula (9) to differentiate $\boldsymbol{W}$ and make the derivative equal to zero:

$$\sum_{i=1}^{d}\frac{\boldsymbol{x}^i\boldsymbol{x}^{iT}\boldsymbol{w}'^i}{\left\|\left[\left(\boldsymbol{w}'^i\right)^T\boldsymbol{x}^i-\boldsymbol{x}^i\right]^T\right\|_{2,1}}+\lambda\sum_{i=1}^{d}\frac{\left(\boldsymbol{w}'^i\right)^T}{\left[\boldsymbol{w}'^i\left(\boldsymbol{w}'^i\right)^T\right]^{\frac{1}{2}}}=\sum_{i=1}^{d}\frac{\boldsymbol{x}^i\boldsymbol{x}^{iT}}{\left\|\left[\left(\boldsymbol{w}'^i\right)^T\boldsymbol{x}^i-\boldsymbol{x}^i\right]^T\right\|_{2,1}}, \tag{11}$$

In formula (11), let:

$$\boldsymbol{D}_1=\sum_{i=1}^{d}\frac{1}{\left\|\left[\left(\boldsymbol{w}'^i\right)^T\boldsymbol{x}^i-\boldsymbol{x}^i\right]^T\right\|_{2,1}}, \tag{12}$$

$$\boldsymbol{D}_2=\sum_{i=1}^{d}\frac{\left(\boldsymbol{w}'^i\right)^T}{\left[\boldsymbol{w}'^i\left(\boldsymbol{w}'^i\right)^T\right]^{\frac{1}{2}}}. \tag{13}$$

Considering that both $\boldsymbol{D}_1\in \mathrm{R}^{n\times n}$ and $\boldsymbol{D}_2\in \mathrm{R}^{d\times d}$ are diagonal matrices, the matrix form of formula (11) can be expressed as:

$$\boldsymbol{X}\boldsymbol{D}_1\boldsymbol{X}^T\boldsymbol{W}'+\lambda\boldsymbol{D}_2\boldsymbol{W}'=\boldsymbol{X}\boldsymbol{D}_1\boldsymbol{X}^T, \tag{14}$$

Simplifying the formula (14), the only optimal $\boldsymbol{W}'$ can be obtained as:

$$\boldsymbol{W}'=\left(\boldsymbol{X}\boldsymbol{D}_1\boldsymbol{X}^T+\lambda\boldsymbol{D}_2\right)^{-1}\left(\boldsymbol{X}\boldsymbol{D}_1\boldsymbol{X}^T\right). \tag{15}$$

However, the complexity of directly calculating $\left(\boldsymbol{X}\boldsymbol{D}_1\boldsymbol{X}^T+\lambda\boldsymbol{D}_2\right)^{-1}$ is high, and the complexity is $O(d^3)$, so in order to improve the calculation efficiency, the generalized inverse lemma of the matrix is used to solve it.

Lemma 1 If the matrix $\boldsymbol{A}\in \mathrm{R}^{n\times n}$ is a non-singular matrix, $\boldsymbol{B}\in \mathrm{R}^{n\times p}$, $\boldsymbol{C}\in \mathrm{R}^{p\times n}$, then:

$$(\boldsymbol{A}+\boldsymbol{B}\boldsymbol{C})^{-1}=\boldsymbol{A}^{-1}-\boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{I}+\boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})^{-1}\boldsymbol{C}\boldsymbol{A}^{-1}. \tag{16}$$

According to formula (14), let $\boldsymbol{A}=\lambda\boldsymbol{D}_2$, $\boldsymbol{B}=\boldsymbol{X}\boldsymbol{D}_1$, $\boldsymbol{C}=\boldsymbol{X}^T$, the new solution form for $\boldsymbol{W}'$ can be obtained as:

$$\boldsymbol{W}'=(\lambda\boldsymbol{D}_2)^{-1}-(\lambda\boldsymbol{D}_2)^{-1}\boldsymbol{X}\boldsymbol{D}_1[\boldsymbol{I}+\boldsymbol{X}^T(\lambda\boldsymbol{D}_2)^{-1}\boldsymbol{X}\boldsymbol{D}_1]\boldsymbol{X}^T(\lambda\boldsymbol{D}_2)^{-1}. \tag{17}$$

Among them, the matrix $\boldsymbol{I}$ is the identity matrix of order n, that is, $\boldsymbol{I}\in \mathrm{R}^{n\times n}$. Formula (17) solves the matrix size of $\boldsymbol{W}'$ smaller than formula (15), so the $\boldsymbol{W}'$ calculated by formula (17) is used for feature selection on the original data to obtain new dimensionality reduction data $\boldsymbol{Y}$.

In the second stage of the $l_{2,p}$-NSPCA algorithm, aiming at the optimization problem (10) at this stage, this chapter constructs the augmented Lagrange function (Augmented Lagrange Multiplier Method, ALMM) model of the target

optimization model, and combines constraints, then optimizes Question (10) would be rewritten as:

$$L(\boldsymbol{W},\boldsymbol{U},\boldsymbol{V},\boldsymbol{\Lambda}) = \min_{r(\boldsymbol{W})=k,\boldsymbol{U}\in\mathbb{R}^{d\times r},\boldsymbol{V}\in\mathbb{R}^{d\times r}} \left\|\left(\boldsymbol{W}^T\boldsymbol{Y}-\boldsymbol{Y}\right)^T\right\|_{2,1} + \alpha\|\boldsymbol{W}\|_{2,1} + \beta(\|\boldsymbol{U}\|_{2,p}^p + \|\boldsymbol{V}\|_{2,p}^p) + <\boldsymbol{\Lambda},\boldsymbol{W}-\boldsymbol{U}\boldsymbol{V}^T> + \frac{\mu}{2}\left\|(\boldsymbol{W}-\boldsymbol{U}\boldsymbol{V}^T)\right\|_F^2. \tag{18}$$

Among them, the Lagrangian multiplier matrix $\boldsymbol{\Lambda}\in\mathbb{R}^{d\times n}$ is introduced, and μ is the Lagrangian multiplier, and the function expression of the formula (18) is the augmented Lagrange form of the $l_{2,p}$-NSPCA algorithm. According to the formula (18), the derivatives of $\boldsymbol{W}$, $\boldsymbol{U}$ and $\boldsymbol{V}$ are respectively performed, and the derivative of $\boldsymbol{W}$ is obtained as follows:

$$\begin{aligned}&\sum_{i=1}^{d}\frac{\boldsymbol{y}^i\boldsymbol{y}^{iT}\boldsymbol{w}^i}{\left\|\left[\left(\boldsymbol{w}^i\right)^T\boldsymbol{y}^i-\boldsymbol{y}^i\right]^{\mathrm{T}}\right\|_{2,1}} + \alpha\sum_{i=1}^{d}\frac{\left(\boldsymbol{w}^i\right)^T}{\left[\boldsymbol{w}^i\left(\boldsymbol{w}^i\right)^T\right]^{\frac{1}{2}}} + \frac{\mu}{2}\left\|\boldsymbol{W}-(\boldsymbol{U}\boldsymbol{V}^T-\frac{\boldsymbol{\Lambda}}{\mu})\right\|_F^2 = 0,\\ &\boldsymbol{W} = \left(\boldsymbol{Y}\boldsymbol{Q}_1\boldsymbol{Y}^T+\alpha\boldsymbol{Q}_2+\mu\boldsymbol{I}\right)^{-1}\left(\mu\boldsymbol{U}\boldsymbol{V}^T-\boldsymbol{\Lambda}\right).\end{aligned} \tag{19}$$

In formula (19), $\boldsymbol{I}\in\mathbb{R}^{d\times d}$ is the unit matrix of order d, and respectively make:

$$\boldsymbol{Q}_1 = \sum_{i=1}^{d}\frac{1}{\left\|\left[\left(\boldsymbol{w}^i\right)^T\boldsymbol{y}^i-\boldsymbol{y}^i\right]^{\mathrm{T}}\right\|_{2,1}}, \tag{20}$$

$$\boldsymbol{Q}_2 = \sum_{i=1}^{d}\frac{\left(\boldsymbol{w}^i\right)^T}{\left[\boldsymbol{w}^i\left(\boldsymbol{w}^i\right)^T\right]^{\frac{1}{2}}}. \tag{21}$$

Both $\boldsymbol{Q}_1\in\mathbb{R}^{n\times n}$ and $\boldsymbol{Q}_2\in\mathbb{R}^{d\times d}$ are diagonal matrices. Derived from formula (18) with respect to $\boldsymbol{U}$:

$$\begin{aligned}&L(\boldsymbol{W},\boldsymbol{U},\boldsymbol{V},\boldsymbol{\Lambda}) = \beta\|\boldsymbol{U}\|_{2,p}^p + \frac{\mu}{2}\left\|\boldsymbol{W}-(\boldsymbol{U}\boldsymbol{V}^T-\frac{\boldsymbol{\Lambda}}{\mu})\right\|_F^2,\\ &\sum_{i=1}^{d}\beta p\boldsymbol{u}^i(\boldsymbol{u}^{iT}\boldsymbol{u}^i)^{\frac{p-2}{2}} - \boldsymbol{Y}^T\boldsymbol{V} - \boldsymbol{W}^T\boldsymbol{V} + \boldsymbol{U}\boldsymbol{V}^T\boldsymbol{V} = 0,\\ &\boldsymbol{U} = \left(\boldsymbol{Y}+\boldsymbol{W}\right)^T\boldsymbol{V}\left(\beta p\boldsymbol{D}-\boldsymbol{V}^T\boldsymbol{V}\right).\end{aligned} \tag{22}$$

make:

$$\boldsymbol{D} = \sum_{i=1}^{d}(\boldsymbol{u}^{iT}\boldsymbol{u}^i)^{\frac{p-2}{2}}. \tag{23}$$

Where $\boldsymbol{D}\in\mathbb{R}^{r\times r}$ is a diagonal matrix, formula (18) can be derived from $\boldsymbol{V}$:

$$\begin{aligned}&L(\boldsymbol{W},\boldsymbol{U},\boldsymbol{V},\boldsymbol{\Lambda}) = \beta\|\boldsymbol{V}\|_{2,p}^p + \frac{\mu}{2}\left\|\boldsymbol{W}-(\boldsymbol{U}\boldsymbol{V}^T-\frac{\boldsymbol{\Lambda}}{\mu})\right\|_F^2,\\ &\boldsymbol{V} = (\mu\boldsymbol{W}^T\boldsymbol{U}-\boldsymbol{\Lambda}^T\boldsymbol{U})\left(\beta p\boldsymbol{D}'+\mu\boldsymbol{U}^T\boldsymbol{U}\right).\end{aligned} \tag{24}$$

Similarly, let:

$$\boldsymbol{D}' = \sum_{i=1}^{d}(\boldsymbol{v}^{iT}\boldsymbol{v}^i)^{\frac{p-2}{2}}. \tag{25}$$

Where $\boldsymbol{D}'\in\mathbb{R}^{r\times r}$ is a diagonal matrix, and the solution to the Lagrangian multiplier term matrix $\boldsymbol{\Lambda}\in\mathbb{R}^{d\times n}$ can be obtained as follows:

$$\boldsymbol{\Lambda} = \boldsymbol{\Lambda}+\mu(\boldsymbol{W}-\boldsymbol{U}\boldsymbol{V}^T). \tag{26}$$

In this chapter, the $l_{2,p}$-NSPCA algorithm is calculated for s iterations, and the ADMM algorithm is used for efficient solution. The specific solution steps of the $l_{2,p}$-NSPCA algorithm are shown in **Algorithm 1**:

Algorithm 1 $l_{2,p}$**-NSPCA Algorithm Steps**

enter: original data matrix $\boldsymbol{X}$, parameter $\lambda>0$, $\alpha>0$ and $\beta>0$, parameter $p>2$, Lagrangian parameters μ

output: matrix $\boldsymbol{U}=\boldsymbol{U}_{k+1}$, $\boldsymbol{V}=\boldsymbol{V}_{k+1}$, $\boldsymbol{\Lambda}=\boldsymbol{\Lambda}_{k+1}$, weight matrix $\boldsymbol{W}$, The data $\boldsymbol{X}$ after the second stage feature selection

step 1. Randomly initialize the first-stage weight matrix $\boldsymbol{W}'\in\mathbb{R}^{d\times d}$, initialization $k=0$, $\boldsymbol{W}_0$, $\boldsymbol{U}_0$ and $\boldsymbol{V}_0$.

step 2. Use formula (12) and formula (13) to calculate the diagonal matrices $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$, respectively.

step 3. Substitute the obtained $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ into formula (17) to obtain $\boldsymbol{W}'$, and obtain the data matrix $\boldsymbol{Y}$ of the first dimension reduction.

step 4. Bring the data matrix $\boldsymbol{Y}$ after the initial dimensionality reduction into the Lagrangian function form (18) of the second-stage optimization model, and calculate the matrix $\boldsymbol{W}_{k+1}=\left(\boldsymbol{Y}\boldsymbol{Q}_1\boldsymbol{Y}^T+\alpha\boldsymbol{Q}_2+\mu\boldsymbol{I}\right)^{-1}\left(\mu\boldsymbol{U}_k\boldsymbol{V}_k^T-\boldsymbol{\Lambda}_k\right)$ according to formula (19).

step 5. Solve the matrix $\boldsymbol{U}_{k+1}=\left(\boldsymbol{Y}+\boldsymbol{W}_{k+1}\right)^T\boldsymbol{V}_k\left(\beta p\boldsymbol{D}-\boldsymbol{V}_k^T\boldsymbol{V}_k\right)$ according to formula (22).

step 6. Solve the matrix $\boldsymbol{V}_{k+1}=(\mu\boldsymbol{W}_{k+1}{}^T\boldsymbol{U}_{k+1}-\boldsymbol{\Lambda}^T\boldsymbol{U}_{k+1})\left(\beta p\boldsymbol{D}'+\mu\boldsymbol{U}_{k+1}{}^T\boldsymbol{U}_{k+1}\right)$ according to formula (24).

step 7. Solve the matrix $\boldsymbol{\Lambda}=\boldsymbol{\Lambda}_k+\mu(\boldsymbol{W}_{k+1}-\boldsymbol{U}_{k+1}\boldsymbol{V}_{k+1}{}^T)$ according to formula (26).

step 8. $k=k+1$.

step 9. According to the formula (10), the feature selection is performed again, and the data matrix $\boldsymbol{Z}$ after the final feature selection is obtained.

4. Experimental Results and Analysis

4.1. Experiment Settings

All experiments were completed on the Windows 10 operating system in Intel Core i5-1135G7 2.4GHz CPU 16GB, using the simulation tool Matlab 2017b. The experiment selected human lung cancer [10] (the human lung carcinomas, LUNG), malignant glioma [10-11] (the malignant glioma, GLIOMA), ALL/AML leukemia data [11] (ALL/AML Leukemia, ALLAML) , colon tumor[11] (Colon Tumor, COLON) and prostate cancer gene expression[10-11] (Prostate Cancer gene expression, PRO-GE) are all high-dimensional gene expression data sets, and the correlation of each data set The properties and their descriptions are shown in Table 1.

Table 1. Relevant properties of the five datasets

Gene dataset name	Sample size	matrix dimension	Number of categories
LUNG	203	3312	5
GLIOMA	50	4434	4
ALLAML	72	3571	2
COLON	62	2000	2
PRO-GE	102	5966	2

4.2. Algorithm Convergence Analysis

The two-stage objective functions of the $l_{2,p}$ -NSPCA algorithm are monotonously decreasing, and since the objective function in the first stage is a convex optimization problem, this section analyzes the convergence of the second stage. Considering that the median value of the regularization parameter adjustment range is 1, set it to 1. The convergence analysis curves of the objective function values of the $l_{2,p}$ -NSPCA algorithm under different data sets are shown in Fig. 1.

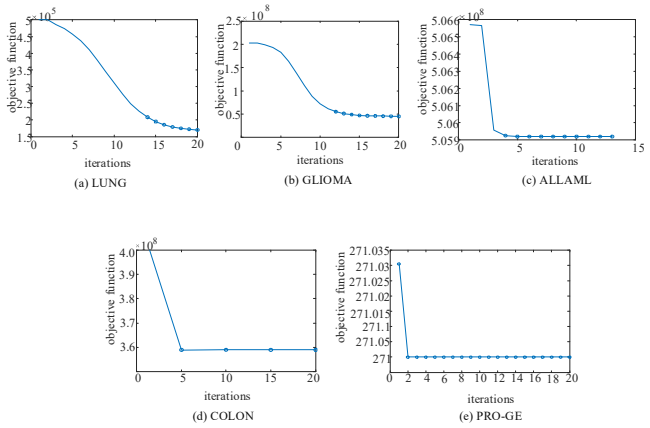

Fig 1. The change results of the objective function value of the $l_{2,p}$ -NSPCA on five data sets

It can be seen from Fig. 1. that the objective function value of the $l_{2,p}$ -NSPCA algorithm in the above five data sets decreases monotonically with the number of iterations, and can quickly converge within 15 iterations on all data sets.

4.3. Clustering Accuracy Analysis

The $l_{2,p}$ -NSPCA algorithm is used in unsupervised feature selection. In order to verify the effectiveness of the $l_{2,p}$ -NSPCA algorithm, the $l_{2,p}$ -NSPCA algorithm is compared with several well-known unsupervised feature selection algorithms. The comparison algorithms are as follows: CSPCA [11-13], Unsupervised Discriminative Feature Selection [11] (Unsupervised Discriminative Feature Selection, UDFS), Multi-Cluster Feature Selection [12] (Multi-Cluster Feature Selection, MCFS), Gaussian Laplacian Algorithm [15] (Laplacian of Gaussian algorithm, LGA), Unsupervised Feature Selection with Multi-Subspace Randomization and Collaboration [16] (Unsupervised Feature Selection with Multi-Subspace Randomization and Collaboration, SRCFS). The K-means clustering algorithm is used to cluster the data obtained after feature selection, and the clustering accuracy (Clustering Accuracy, ACC) is used as an index for evaluating the performance of the feature selection algorithm. Let q_i denote the cluster label of a clustering algorithm, and p_i denote the true label value of the data matrix $\boldsymbol{X}$. ACC is defined as follows[18]:

$$ACC = \frac{\sum_{i=1}^{n} \delta\left(p_i, \mathrm{map}\left(q_i\right)\right)}{n}, \tag{27}$$

Here, $\mathrm{d}(x, y)$ denotes the indicator function, if $x = y$, $\mathrm{d}(x, y) = 1$, otherwise $\mathrm{d}(x, y) = 0$. The larger the ACC, the better the clustering effect.

In the experiment, each group of data was randomly repeated clustering 30 times, and the best clustering accuracy was selected as the final clustering accuracy. All algorithm parameters in the experiments will be selected in the set $\{10^{-6}, 10^{-4}, 10^{-2}, 10^{0}, 10^{2}, 10^{4}, 10^{6}\}$, with 20% and 40% feature selection for the datasets in Table 1, respectively. When 20% features are selected, the optimal clustering accuracy of the six algorithms are shown in Table 2. The $l_{2,p}$ -NSPCA algorithm selects 80% of the features in the first stage, and 25% of the features should be selected in the second stage to ensure that the final selected feature range is 20%.

Table 2. The optimal clustering accuracy of the six algorithms when feature selection is 20% (%)

data set	UDFS	MCFS	LGA	SRCFS	CSPCA	$l_{2,p}$ -NSPCA
LUNG	69.54	69.73	69.49	72.09	70.05	**75.24**
GLIOMA	53.93	53.67	59.2	56.47	56.73	**59.35**
ALLAML	89.77	88.06	91.06	93.1	91.11	**93.20**
COLON	54.57	53.55	52.26	53.23	56.34	**57.94**
PRO-GE	53.25	56.82	57.42	57.45	57.65	**59.29**

When 40% is selected, the comparison results of the optimal clustering accuracy of the six algorithms on the five gene expression datasets are shown in Table 3. The $l_{2,p}$ -NSPCA algorithm selects 80% of the features in the first stage and 50% of the features in the second stage, ensuring that the final selected feature range is 40%.

It can be seen from Table 2 and Table 3 that when the feature selection range of the data is 20% and 40%, among the six typical algorithms, The clustering accuracy of $l_{2,p}$ -NSPCA algorithm is the best.

4.4. Operational Efficiency Analysis

In order to verify the running time comparison of $l_{2,p}$ -NSPCA algorithm compared with CSPCA and other algorithms, 20% and 40% of the data characteristics were

selected respectively. The running time of the six related comparison algorithms under the optimal precision is shown in Table 4 and Table 5 respectively.

Table 3. The optimal clustering accuracy of the six algorithms when feature selection is 40% (%)

data set	UDFS	MCFS	LGA	SRCFS	CSPCA	$l_{2,p}$-NSPCA
LUNG	65.75	68.58	72.53	70.74	72.33	**77.35**
GLIOMA	56.73	58.4	57.27	57.4	57.67	**58.6**
ALLAML	83.61	87.92	89.91	91.39	90.79	**92.08**
COLON	53.71	53.92	52.53	53.49	54.19	**57.6**
PRO-GE	52.22	57.52	57.27	58.46	58.67	**59.98**

Table 4. The running time (s) corresponding to the optimal accuracy of the six algorithms when selection is 20%

data set	UDFS	MCFS	LGA	SRCFS	CSPCA	$l_{2,p}$-NSPCA
LUNG	58.0964	260.7039	2.2645	1.7798	314.9214	205.6568
GLIOMA	143.2086	584.0475	0.9075	0.4204	321.7139	224.2786
ALLAML	75.334	301.1364	1.0962	0.3926	367.5574	140.6905
COLON	11.9014	55.9812	1.2940	0.6335	58.3943	11.3927
PRO-GE	405.0692	878.4173	0.4032	2.5209	135.2105	80.7374

Table 5. When the feature selection of the data set is 40%, the running time (s) corresponding to the optimal accuracy of the six algorithms

data set	UDFS	MCFS	LGA	SRCFS	CSPCA	$l_{2,p}$-NSPCA
LUNG	62.3085	256.2922	0.6343	1.2005	68.2137	37.1454
GLIOMA	153.9155	753.3435	0.373	0.5158	873.1727	342.0899
ALLAML	85.5898	310.7413	0.3159	1.4741	428.0371	76.4352
COLON	13.1968	53.6874	0.3869	0.4291	14.9156	10.9015
PRO-GE	460.5854	950.9047	0.5667	0.7097	103.265	30.4928

Table 4 and Table 5 above record the running time of the six algorithms at optimal accuracy when the data features are selected at 20% and 40% respectively. It can be seen from Table 4 and Table 5 that when the feature selection range is 20% and 40%, compared with the original CSPCA algorithm, the $l_{2,p}$-NSPCA algorithm reduces the overall calculation running time, and when the feature selection range is 40%, $l_{2,p}$-NSPCA's The overall running time is less than UDFS and MCFS algorithms. In some data sets, the running complexity of $l_{2,p}$-NSPCA is lower than that of UDFS and MCFS algorithms.

5. Conclusion and Future Work

This chapter proposes a non-convex sparse principal component analysis: $l_{2,p}$-NSPCA algorithm, and the proposed $l_{2,p}$-NSPCA algorithm is applied to unsupervised feature selection, $l_{2,p}$-NSPCA is a two-stage algorithm, in which the first stage performs a sparse PCA algorithm to perform initial feature selection on the data, in which the generalization of the matrix is introduced Inverse lemma to reduce the complexity of solving the algorithm. The second stage of the $l_{2,p}$-NSPCA algorithm executes the SPCA algorithm with a low-rank penalty item, so as to perform feature selection and dimensionality reduction processing on the data that has been feature-selected in the first stage. Among them, for the matrix low-rank regularization term in the second stage, this chapter first performs matrix decomposition on the weight matrix, and then uses $l_{2,p}$ - norm $(2 < p < +\infty)$ to replace the original kernel norm regularization term, making it a non-convex relaxed low-rank regularization term. Rank approximation penalty term.

The comparative experimental results on five real data sets show that $l_{2,p}$-NSPCA algorithm not only outperforms the original CSPCA algorithm in feature selection performance, but also shows advantages in running speed.

References

[1] Nie F, Huang H, Ding C. Low-rank matrix recovery via efficient schatten p-norm minimization[C]. Proceedings of the AAAI Conference on Artificial Intelligence. 2017, 26(1): 655-661.

[2] Wang J, Xie F, Nie F, et al. Unsupervised Adaptive Embedding for Dimensionality Reduction[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, PP (99): 1-12.

[3] Zhao H, Wang Z, Nie F. A New Formulation of Linear Discriminant Analysis for Robust Dimensionality Reduction[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(4): 629-640.

[4] Ang J C, Mirzal A, Haron H, et al. Supervised, Unsupervised, and Semi-Supervised Feature Selection: A Review on Gene Selection[J]. IEEE/ACM Transactions on Computational Biology & Bioinformatics, 2016, 13(5): 971-989.

[5] Davenport M A, Romberg J. An overview of low-rank matrix recovery from incomplete observations[J]. IEEE Journal of Selected Topics in Signal Processing, 2016, 10(4): 608-622.

[6] Candès E J, Li X, Ma Y, et al. Robust principal component analysis? [J]. Journal of the ACM (JACM), 2011, 58(3): 1-37.

[7] Fan J, Ding L, Chen Y, et al. Factor group-sparse regularization for efficient low-rank matrix recovery[J]. Advances in Neural Information Processing Systems, 2019, 32(14): 78-103.

[8] Chang X, Nie F, Yang Y, et al. Convex sparse PCA for unsupervised feature learning[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2016, 11(1): 1-16.

[9] Giampouras P V, Rontogiannis A A, Koutroumbas K D. Robust PCA via alternating iteratively reweighted low-rank matrix factorization[C]. The 2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018: 3383-3387.

[10] Nie F, Hu Z, Li X. Matrix completion based on non-convex low-rank approximation[J]. IEEE Transactions on Image Processing, 2018, 28(5): 2378-2388.

[11] Chi Y, Lu Y M, Chen Y. Nonconvex optimization meets low-rank matrix factorization: An overview[J]. IEEE Transactions on Signal Processing, 2019, 67(20): 5239-5269.

[12] Sun R, Luo Z Q. Guaranteed matrix completion via non-convex factorization[J]. IEEE Transactions on Information Theory, 2016, 62(11): 6535-6579.

[13] Aftab K, Hartley R. Convergence of iteratively re-weighted least squares to robust m-estimators[C]. The 2015 IEEE Winter Conference on Applications of Computer Vision. IEEE, 2015: 480-487.

[14] Abdi H, Williams L J. Principal component analysis[J]. Wiley interdisciplinary reviews: computational statistics, 2010, 2(4): 433-459.

[15] Zhi Xiaobin, Li Yalan. Two-stage Discriminant Embedded Clustering [J]. Journal of Xi'an University of Posts and Telecommunications, 2018, 23(03): 45-51.

[16] Recht B, Fazel M, Parrilo P A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization[J]. SIAM review, 2010, 52(3): 471-501.

[17] Wright J, Ganesh A, Rao S, et al. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization[J]. Advances in neural information processing systems, 2009, 22: 17-30.

[18] Ornhag M V, Olsson C. A unified optimization framework for low-rank inducing penalties[C]. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 8474-8483.

[19] Zhao T, Wang Z, Liu H. A nonconvex optimization framework for low rank matrix estimation[J]. Advances in Neural Information Processing Systems, 2015, 28(3): 8-21.

[20] Mo D, Lai Z. Robust Jointly Sparse Regression with Generalized Orthogonal Learning for Image Feature Selection[J]. Pattern Recognition, 2019, 93: 164-178.