A Robust Linear Regression Feature Selection Method for Data Sets with Unknown Noise

Yaqing Guo, Wenjian Wang, and Xuejun Wang

Abstract—The linear regression model is simple in form and easy to estimate; nevertheless, irrelevant features will raise the difficulty of its tasks. Feature selection is generally adopted to improve the model performance. Unfortunately, traditional regression feature selection methods may not work for data with noise or outliers. Although some robust methods for certain specific error distributions have been proposed, they may not perform well because the distribution of representation error is often unknown for real data. This paper proposes a regression feature selection method for unknown noise named Mixture of Gaussians LASSO (MoG-LASSO), in which feature selection and model training will be achieved simultaneously. MoG is adopted to model unknown noises, and M-estimation is used to acquire the weighted squared error loss. By alternatively and iteratively updating the regression coefficient and parameters of MoG, the influence of unknown noise can be reduced effectively. Furthermore, MoG-LASSO achieves feature selection by the L_1 regularization term, which can further improve the performance of the model. Experimental results on artificial data and benchmark data sets demonstrate that MoG-LASSO has better robustness and sparsity for data sets with irrelevant features. Additionally, experimental results on face recognition databases show the performance advantage of MoG-LASSO over state-of-the-art methods in the presence of illumination variations.

Index Terms—Regression, feature selection, unknown noise, robust, face recognition.

1 Introduction

Linear regression is a common model for data fitting and prediction, and it is simple in form and easy to estimate. For some actual tasks, such as bioinformatics mining, text classification and image retrieval, data often contain a large number of features, in which many irrelevant features may be embedded. This will raise the difficulty of linear regression modeling, which may lead to bad learning effects and model interpretability. In addition, observation of some features is often expensive. If these are irrelevant features, many unnecessary costs will be incurred. Feature selection is proposed to address these challenging tasks involving many irrelevant features. It is a process that selects irrelevant features from data sets [1], [2].

At present, the representative feature selection methods for regression problems can be divided into two categories. The first assesses subsets of features according to their usefulness to a given predictor. They finish feature selection first and then train the model. Representative methods are Best-Subset Selection, Forward-Stepwise Selection, Backward-Stepwise Selection, Stepwise Regression Method, etc. [3]. Best-subset selection acquires optimal feature sets by evaluating each subset according to a certain criterion, such as the mean squared error respect to the difference between the sample size and feature subset size, the C_p statistic and AIC. Forward-stepwise selection, backward-stepwise selection and stepwise regression method are greedy search strategies. When a certain feature is introduced or removed, they calculate and compare the F test statistic of each feature [4]. The above methods consider learner performance during feature selection, but their feature selection may be extremely variable even the noises are slight due to their inherent discreteness. Moreover, stepwise feature selection can be easily trapped into the local optimal solution. These methods are not suitable for data sets that have many features, so they are not commonly used methods. Another type of method named the regularized method finishes feature selection and learner training simultaneously, which acquires sparse solutions by regularization terms. Differently with the above feature selection methods, they are applied in many areas, such as classification, face recognition and compressed sensing [2]. In addition, regularized methods are also used purely as a good feature selection method for better classification. For example, MRM-LASSO selects the most discriminative features from multiple views simultaneously under pattern-specific weights, which can indicate the contribution of each pattern to labels. Then it fuses classification results based on each view-specific relevant features as the final prediction result [5].

Existing regression feature selection methods may not work well in most real problems where noise is unknown and more complicated than specific error distributions. For instance, in face recognition databases, face images that are taken under varying illumination conditions of one subject contain cast shadows, specular reflections or camera noises [6]. These different types of noise in face images follow different distributions. Hence, identifying above noisy face images which can be represented as response vectors will introduce larger energy to linear regression feature selection analysis based face recognition models. MoG (the mixture of Gaussians distribution) is a universal approximator of any

• *The authors are with the Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan 030006, Shanxi Province, China.*
E-mail: {791771653, 365690064}@qq.com, wjwang@sxu.edu.cn.

Manuscript received - -, -; revised - -, -.
(Corresponding author: Wenjian Wang.)

continuous distribution and many discontinuous distributions. Recently, Ref. [7] used MoG to model unknown noise in matrix factorization and obtained the loss function that fits noises according to M-estimation.

Motivated by the work of Ref. [7], this paper proposes a robust linear regression feature selection method named MoG-LASSO for unknown noise. It selects MoG to model unknown noise and then acquires a loss function that corresponds to noises of data sets according to M-estimation. MoG-LASSO updates parameters of MoG iteratively according to updated regression coefficients and the EM algorithm, and better weights of samples will be acquired during the procedure of MoG-LASSO. It finishes feature selection and model training simultaneously to improve learner performance and simplify linear regression models. In so doing, the influence of unknown noise can be better reduced. Experimental results on both artificial data and benchmark data sets demonstrate that the proposed MoG-LASSO has better robustness and sparsity for data sets with irrelevant features. In addition, MoG-LASSO has been verified on face recognition databases, and the experimental results show the performance advantage of MoG-LASSO over the state-of-the-art methods in the presence of illumination variations.

The rest of this paper is organized as follows. Section 2 reviews existing regularized regression feature selection methods. Section 3 shows some preliminaries, including M-estimation, several classical regression methods and sparse representation based classification. Section 4 explains the idea of the proposed MoG-LASSO in detail and provides a solution for model optimization. Experimental results are presented and analyzed in Section 5. Finally, we conclude this paper in Section 6.

2 RELATED WORK

In this section, we briefly review existing regularized regression feature selection methods.

There are currently various regularized methods used in linear regression modeling. Ridge regression with the L_2 regularization term is a typical regularized method, but its solutions are generally not sparse because of the L_2 regularizer [8]. Tibshirani proposed the famous LASSO model, which uses an L_1 regularization term that can easily acquire sparse solutions to replace the L_2 regularization term of ridge regression [9]. The efficient solving algorithm for LASSO named Least Angle RegresSion (LARS) was proposed in 2004 [10], and then the coordinate descent method was used to further improve its efficiency [11]. LASSO has aroused broad interest since it was proposed because of its sparsity. However, LASSO does not always work well in terms of feature selection and estimation of regression coefficients. Some regularized methods have been proposed to solve these problems through useful regularization terms.

In the aspect of feature selection, LASSO has some limitations, such as not being able to produce the sparsest solution, obtain a better feature selection result when noises follow a heavy tail distribution, or handle highly correlated features and structurally grouped features very well. Some improved models have been proposed to solve the above problems. $L_{1/2}$ regularization was proposed to replace the L_1 regularization term [12], and it can acquire better feature selection results than LASSO even when data sets contain heavy tail noises. However, the thresholding estimates of the $L_{1/2}$ regularizer do not satisfy continuity, which is argued by Fan et al. that a good penalty function should satisfy to avoid instability of model prediction in data sets with noises [13]. The elastic net, which makes a compromise between the ridge and the LASSO penalties, was proposed to handle highly correlated features [14]. Its coefficient paths do not tend to be erratic or show wild behavior like LASSO. Therefore, it can pull highly correlated features together and reflect the relative importance of the individual features, and its prediction accuracy is also better than that of LASSO. The elastic net does not consider the fact that different degrees of correlation may exist among different features, e.g., there may be no correlation among some features in many situations. Therefore, the ridge penalty may cause biased estimation. Group LASSO was proposed to solve regression problems in which the covariates have a natural group structure [15]. It can acquire the desirable result that all coefficients within a group become irrelevant features (or relevant features) simultaneously. The grouping situation of Group LASSO needs to be preestablished artificially, and its estimation is biased. The $l_{1,2}$ norm feature selection which can select features that are most correlated for each class separately is a special case of Group LASSO [16].

LASSO gives all components of regression coefficients the same degree of punishment. Not only are components of a regression coefficient about irrelevant features set to 0, but components of relevant features are also given a certain degree of compression. This may lead to a biased estimation of regression coefficients. To solve this problem, sparse models that are approximately unbiased, such as Smoothly Clipped Absolute Deviation (SCAD) [17] and Minimax Concave Penalty (MCP) [18], have been proposed. These methods compress each component of regression coefficients to different degrees. SCAD and MCP can only slightly reduce the generalization error of LASSO. And differently with LASSO, their regularization terms are non-convex, which leads to the inexistence of global optimal solutions and difficulty for optimization. Hence, LASSO is still widely used due to its quick solution and acceptable feature selection results.

The loss functions of the above regularized methods are the squared error loss. When noises embedded in data sets are nonnormal noises, the above regularized methods with squared error loss may not be robust and sparse. To solve this problem, some robust regularized methods, the maximum correntropy induced loss [19], the absolute loss (LAD-LASSO [20], LAD-Adaptive LASSO [21], etc.) and the quantile loss (quantile regression LASSO and its improved methods [22], [23], [24], which are often complicated) have been proposed. Among them, the maximum correntropy induced loss can handle non-zero mean noises, and its estimation performance depends strongly on the kernel scale parameter. However, the choice of a suitable kernel scale parameter for model learning is very difficult. All of the above remaining robust methods are M-estimation, which is a generalized maximum likelihood method proposed by Huber [25]. When noise follows different distributions, the M-estimator will correspond to different loss functions, i.e.,

the squared loss function (Gaussian), the absolute loss function (Laplace) and the quantile loss function (asymmetric Laplace). According to M-estimation, these methods are robust for corresponding error distributions, and they can acquire good results of feature selection and model training [26].

However, provided a priori knowledge of the error distribution, which is typically not available, and a regression feature selection method could not be selected in mind efficiently. Fortunately, the asymmetric exponential power distribution (AEPD) holds the Gaussian, Laplace, asymmetric Laplace, etc., as special cases when its parameters take different values. Ref. [26] proposed a robust regression methodology named the adaptive M-estimator (AME) that corresponds to M-estimation when the noises obey the AEPD. It can select the best loss function from a broad class in a data-driven fashion, but it cannot finish feature selection. In addition, its AEPD can only model certain error distributions, and it is limited to unknown noise in the real world.

The above regression feature selection methods are extremely sensitive to outliers that deviate from the model assumptions. Based on robust regression methodologies for outliers such as WLAD (Weight Least Absolute Deviation) [27] and LST (Least Trimmed Squares estimator) [28], WLAD-LASSO [27], LTS-LASSO [28], reweighted LTS-LASSO [28], WLAD-CATREG (categorical regression model) adoptive elastic net [29], WLAD-SCAD [30], etc. are proposed. Among them, LTS-LASSO reduces the influence of outliers by using a subset with smaller training error of data sets as the training set, but it is often computationally expensive. The remaining methods for outliers increase their robustness by weighting loss functions of samples. Reweighted LTS-LASSO regards the regression coefficient from LTS-LASSO as an initial parameter value. The weights of other weighted methods will be computed by using robust location, scatter estimators and robust distances of data sets. For the above weighted methods, weights are set in advance and fixed throughout feature selection and model training, which leads to poor adaptability.

3 PRELIMINARIES

To illustrate the proposed method clearly, M-estimation, several classical regression coefficient estimation methods and classification based sparse representation (SRC) are introduced briefly in this section. Among them, the classification process of SRC is selecting training samples which belong to the same class with the test sample in nature. And this process can be regard as feature selection. Hence, the proposed robust linear regression feature selection method can also solve classification problems by combining with SRC.

3.1 M-estimation

As a generalized maximum likelihood method, M-estimation induces estimators that estimate the regression coefficient under robust loss functions. The data set $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is given, where $\mathbf{x}_i \in R^p$ and $y_i \in R$. Consider the linear regression model $y_i = \mathbf{x}_i^T\boldsymbol{\beta} + \varepsilon_i$, where $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots, \beta_p)^T \in R^p$ is the regression coefficient and $\varepsilon_i \in R$ are the iid random noises with a known probability density function f. Therefore, the likelihood function is $L(\boldsymbol{\beta}) = \prod_{i=1}^n f(y_i - \mathbf{x}_i^T\boldsymbol{\beta})$, and the log-likelihood function is $\ln L(\boldsymbol{\beta}) = \sum_{i=1}^n \ln f(y_i - \mathbf{x}_i^T\boldsymbol{\beta})$. From M-estimation, the optimum of the regression coefficient $\boldsymbol{\beta}$ under f can be acquired by maximizing $\ln L(\boldsymbol{\beta})$. Maximizing the log-likelihood function implies that the loss function $S(y_i - \mathbf{x}_i^T\boldsymbol{\beta})$ needs to be minimized, i.e., optimization of the loss function $S(y_i - \mathbf{x}_i^T\boldsymbol{\beta})$ can obtain the optimum of $\boldsymbol{\beta}$. Therefore, one can get the loss function corresponding to f according to M-estimation [25], [26]. For example, ε_i is a noise sample from the Gaussian distribution $N(0, \sigma^2)$, and the log-likelihood function is $\ln L(\boldsymbol{\beta}) = n\ln\frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2}\sum_{i=1}^n (y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2$. Maximizing the log-likelihood function requires minimizing the squared error loss function. Models can acquire better results for both feature selection and learning performance for Gaussian noise when their loss functions are the squared error loss.

3.2 Classical Regression Coefficient Estimation Methods

Several classical and competitive linear regression models will be introduced in this subsection, and they will be compared with the proposed model in experiments.

For the linear regression model $y_i = \mathbf{x}_i^T\boldsymbol{\beta} + \varepsilon_i$, the Least Squares Estimator (LSE) is the most popular estimation approach, which is an M-estimator [27]. The optimization problem of LSE is:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^n (y_i - \sum_{j=1}^p x_{ij}\beta_j)^2. \tag{1}$$

It is well-known that LSE needs to find the inversion of $\mathbf{X}^T\mathbf{X}$ when solving the above problem, where $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n)^T$ is the feature matrix. However, in fact, $\mathbf{X}^T\mathbf{X}$ may not be invertible (i.e., $\mathbf{X}^T\mathbf{X}$ is the singular matrix) in some situations, such as the feature size being larger than the sample size. In addition, LSE easily becomes stuck in overfitting when the feature size is larger and the sample size is smaller.

LASSO with the L_1 regularizer can avoid matrix inversion and relieve overfitting effectively, and the model is as follows:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^n (y_i - \sum_{j=1}^p x_{ij}\beta_j)^2 + \lambda\|\boldsymbol{\beta}\|_1, \tag{2}$$

where $\lambda > 0$ is the regularization parameter. Compared with LSE, LASSO can improve its prediction accuracy through feature selection. The optimization problem (2) can be solved by Least Angle RegresSion (LARS) [10], Homotopy [31] and the coordinate descent method [11], etc.

LADE is less sensitive to heavy-tailed noises, such as Laplace noise, and its optimization objective function is [27]:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^n |y_i - \sum_{j=1}^p x_{ij}\beta_j|. \tag{3}$$

LAD-LASSO can finish feature selection and improve prediction accuracy, and the regression parameters are estimated by minimizing the following objective function:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} |y_i - \sum_{j=1}^{p} x_{ij}\beta_j| + \lambda \|\boldsymbol{\beta}\|_1. \tag{4}$$

It is solved by transformation into a linear programming problem [32].

Subspace-MoG has good robustness when matrix factorization data sets contain unknown noises [7]. Its loss function that fits noises well is obtained according to M-estimation, and its model is:

$$\min_{\mathbf{U},\mathbf{V}} \|\mathbf{W} \odot (\mathbf{X} - \mathbf{U}\mathbf{V}^T)\|_{L_2}, \tag{5}$$

where $\mathbf{X} \in R^{d\times n}$, $\mathbf{U} \in R^{d\times r}$ and $\mathbf{V} \in R^{n\times r}$ are low-dimensional matrices ($r < \min(d, n)$). The element w_{ij} of $\mathbf{W} \in R^{d\times n}$ is

$$w_{ij} = \begin{cases} \sqrt{\sum_{k=1}^{K} \frac{\gamma_{ijk}}{2\sigma_k^2}}, & i,j \in \Omega \\ 0, & i,j \notin \Omega \end{cases}$$

and γ_{ijk} and σ_k are the responsibility and parameters from MoG, respectively. For the matrix $\mathbf{D}$ whose element is d_{ij}, $\|\mathbf{D}\|_{L_p} = \sum_{i,j} |d_{ij}|^p$ is the power p norm of $\mathbf{D}$. In addition, we believe that the loss function of Subspace-MoG can be applied in linear regression problems by the following formulation: $\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} w_i (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2$.

3.3 Classification Based on Sparse Representation

Sparse representation [33] is an effective tool in a myriad of applications, such as classification, face recognition, and compressed sensing. For classification problems, any test sample belonged to the $c-$th class will approximately lie in the linear span of the training samples from the same class for some scalars when sufficient training samples associated with object c are given. Considering the entire training set, the liner coefficient vector of the test sample is a vector whose entries are zero except those associated with the $c-$th class. Naturally, classification based on sparse representation (SRC) aims to solve the following minimization problem:

$$\mathbf{x} = \arg\min_{\mathbf{x}} \|\mathbf{Y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \parallel \mathbf{x} \parallel_0 . \tag{6}$$

Where $\mathbf{Y} \in R^m$ is a test sample, $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \cdots, \mathbf{A}_C] \in R^{m\times n}$ is the training set that contains C classes, $A_c = [\mathbf{v}_{c,1}, \mathbf{v}_{c,2}, \cdots, \mathbf{v}_{c,n_c}] \in R^{m\times n_c}$ is the $c-$th class, $\mathbf{v}_{c,i}$ is a sample of the $c-$th class, $i = 1, 2, \cdots, n_c$, and $n = \sum_{c=1}^{C} n_c$. After optimizing equation (6), the sparse representation of the test sample Y in the training set A with C classes can be acquired, i.e., $\mathbf{x}_Y = [0, \cdots, 0, \iota_{c,1}, \iota_{c,2}, \cdots, \iota_{c,n_c}, 0, \cdots, 0]^T \in R^n$, where $\iota_{c,i}$ is the scalar of the training sample $\mathbf{v}_{c,i}$. Hence, the test sample Y can be better represented by samples of the c class, and then it is classified as class c.

Unfortunately, the minimization problem of Eq. (6) contained the L_0 norm that simply counts the number of nonzero entries in $\mathbf{x}$ is NP hard, and may be computationally infeasible for large-scale problems. It has been proven that, the L_1 norm $\| \bullet \|_1 = \sum_j |x_j|$ minimization is equivalent to the L_0 norm minimization with high probability under certain conditions on $\mathbf{A}$. And Yang et al. argued that the L_1 regularizer is more suitable than L_0 for the framework of SRC and gave some theoretical support for their viewpoint. They thought that the L_0 regularizer does not own closeness, which makes the nonzero representation coefficients concentrating on the training samples belong to the same class with the test sample [34]. Thus, one can seek the desired $\mathbf{x}$ by solving the following convex optimization problem [33]:

$$\mathbf{x} = \arg\min_{\mathbf{x}} \|\mathbf{Y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \parallel \mathbf{x} \parallel_1 . \tag{7}$$

The above optimization problem is also the model of LASSO. Then, the test sample can be classified based on the following rule: $Label(\mathbf{Y}) = \arg\min_c r_c(\mathbf{Y})$, where $r_c(\mathbf{Y}) = \|\mathbf{Y} - \mathbf{A}_c\mathbf{x}_c\|_2$.

Comparing with the linear regression classifier (LRC) [35] which acquires the linear representation only based on one class, SRC with a global representation can avoid over-fitting and achieve classification even classes are highly correlated to each other. Moreover, it can recognize and reject outlying samples that do not belong to any class according to the sparse representation.

4 MOG-LASSO

In this section, we will explain the proposed MoG-LASSO method in detail. MoG is used to model unknown noises, and then the MoG-LASSO model is acquired by M-estimation and the L_1 regularization term. The solution for optimizing the model is also provided. Finally, the MoG-LASSO algorithm is described.

4.1 Unknown Noise Modeling

According to M-estimation theory, there is a corresponding relationship between the form of the loss function and the noise of the data sets. The robust problem will appear if simple forms of loss functions underfit unknown noise of data sets. That is, the absolute or squared loss function will acquire a good regression coefficient when noises of data sets follow a Laplace or Gaussian distribution. In addition, in Bayesian linear regression, the likelihood function, which represents the training data probability under the unknown regression coefficient, corresponds exactly to the loss function in regression model learning. The randomness of the training data comes from noise. Hence, from the Bayesian perspective, the same robust problem will also occur when loss functions do not match the distribution of unknown noises. For unknown noises, such as multi-modal distribution noises, MoG may be an alternative choice. It is a universal approximator to any continuous distribution and many discontinuous distributions [36]. Here, MoG is selected to model unknown noise and derive the loss function that matches noises contained in data sets through M-estimation.

Assume that ε_i is a sample from a MoG distribution $p(\varepsilon)$, i.e., $p(\varepsilon) \sim \sum_{k=1}^{K} \pi_k N(\boldsymbol{\varepsilon}|0, \sigma_k^2)$, where $N(\boldsymbol{\varepsilon}|0, \sigma_k^2)$ represents

the Gaussian distribution whose mean is 0 and variance is σ_k^2. It is the $k-$th component of MoG. $\pi_k \geq 0$ is the mixing proportion, and $\sum_{k=1}^{K} \pi_k = 1$. The number of Gaussians K is relative to the complexity of unknown noises in data sets, and MoG-LASSO estimates it according to Ref. [7]. First, K is set to a sufficiently large fixed value (e.g., $K = 6$) for fitting the noise distribution, and then its value is adjusted according to the reality of the data sets and the following criterion. If the relative deviation $\frac{|\sigma_i^2 - \sigma_j^2|}{\sigma_i^2 + \sigma_j^2}$ between variances of the $i-$th and $j-$th Gaussian is smaller than the threshold τ, then

$$\pi_i = \pi_i + \pi_j, \tag{8}$$

$$\sigma_i^2 = \frac{n_i \sigma_i^2 + n_j \sigma_j^2}{n_i + n_j}, \tag{9}$$

$$K = K - 1.$$

Where n_i and n_j represent the number of samples in the $i-$th and $j-$th Gaussian, respectively.

Then, the probability of each response variable y_i of $\mathbf{Y}$ can be written as $p(y_i | \sum_{j=1}^{p} x_{ij}\beta_j, \Pi, \Sigma)$, where $\Pi = \{\pi_1, \pi_2, \cdots, \pi_K\}$ and $\Sigma = \{\sigma_1, \sigma_2, \cdots, \sigma_K\}$. The likelihood of $\mathbf{Y}$ is

$$p(\mathbf{Y}|\boldsymbol{\beta}, \Pi, \Sigma) = \prod_{i=1}^{n} p(y_i | \sum_{j=1}^{p} x_{ij}\beta_j, \Pi, \Sigma).$$

One can acquire the log-likelihood function of $\mathbf{Y}$ by taking the logarithm of the above likelihood, that is:

$$\max_{\boldsymbol{\beta}, \Pi, \Sigma} L(\boldsymbol{\beta}, \Pi, \Sigma) = \sum_{i=1}^{n} \log p(y_i | \sum_{j=1}^{p} x_{ij}\beta_j, \Pi, \Sigma). \tag{10}$$

By maximizing the above objective function, the loss function form and parameters of MoG that correspond to unknown noise can be obtained.

4.2 MoG-LASSO Model

To solve the optimization problem (10), we need to know which component of the MoG that ε_i comes from. Therefore, the EM algorithm [37] is adopted to optimize the log-likelihood function of $\mathbf{Y}$. The EM algorithm is the maximum likelihood estimation for probabilistic models with latent variables, and it achieves maximization of the log-likelihood function by iteratively maximizing the lower bound of the log-likelihood function. The lower bound is transformed into the Q function in the EM algorithm.

Calculate the responsibilities γ_{ik} of ε_i to the $k-$th component in step E, and finish determining the form of the loss function and estimating the parameters of MoG and the regression coefficient $\boldsymbol{\beta}$ in step M. $\varepsilon_i = y_i - \sum_{j=1}^{p} x_{ij}\beta_j$ is used to represent noise from a mixture of Gaussians.

E Step: Assume the latent variable in the model is z_{ik}, with $z_{ik} = \{0, 1\}$, $\sum_{k=1}^{K} z_{ik} = 1$, and

$$z_{ik} = \begin{cases} 0 & \varepsilon_i \text{ comes from the } k-\text{th component} \\ 1 & \text{otherwise.} \end{cases}$$

The solution formula for the posterior responsibility of the $k-$th component that generates the noise ε_i is:

$$E(z_{ik}) = \gamma_{ik} = \frac{\pi_k N(\varepsilon_i | \sum_{j=1}^{p} x_{ij}\beta_j, \sigma_k^2)}{\sum_{k=1}^{K} \pi_k N(\varepsilon_i | \sum_{j=1}^{p} x_{ij}\beta_j, \sigma_k^2)}, \tag{11}$$

where $i = 1, 2, \cdots, n$ and $k = 1, 2, \cdots, K$.

According to the log-likelihood function and responsibility formula, the Q function can be acquired after some derivations, that is:

$$Q : \sum_{k=1}^{K} \sum_{i=1}^{n} r_{ik} (\ln \pi_k - \ln \sqrt{2\pi}\sigma_k - \frac{(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2}{2\sigma_k^2}).$$

Maximizing the Q function can maximize the log-likelihood function. Then, in the M step, the MoG-LASSO model is decided in the process of optimization the Q function.

M Step (Π and Σ): Finish estimating parameters of MoG by maximizing the Q function. The closed-form of the MoG parameters is as follows:

$$\pi_k = \frac{\sum_{i=1}^{n} r_{ik}}{n}, \tag{12}$$

$$\sigma_k^2 = \frac{\sum_{i=1}^{n} r_{ik} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2}{\sum_{i=1}^{n} r_{ik}}. \tag{13}$$

M Step ($\boldsymbol{\beta}$): The components of the Q function related to $\boldsymbol{\beta}$ can be rewritten as follows:

$$\begin{aligned} &\sum_{k=1}^{K} \sum_{i=1}^{n} r_{ik} (-\frac{(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2}{2\sigma_k^2}) \\ &= -\sum_{i=1}^{n} (\sum_{k=1}^{K} \frac{\gamma_{ik}}{2\sigma_k^2}) (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2. \end{aligned} \tag{14}$$

It is easy to observe that maximizing the log-likelihood function of MoG noises requires minimizing the weighted squared loss function. Therefore, according to M-estimation and Bayesian, the form of the loss function should be the weighted squared loss when data sets contain unknown noise, i.e.,

$$\sum_{i=1}^{n} w_i (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2,$$

where the weights of the loss function are $w_i = \sum_{k=1}^{K} \frac{\gamma_{ik}}{2\sigma_k^2}$.

Optimizing the weighted squared loss function requires the inversion of $\mathbf{X}^T\mathbf{W}\mathbf{X}$, where $\mathbf{W}$ is the diagonal weight matrix. To avoid finding the inversion of $\mathbf{X}^T\mathbf{W}\mathbf{X}$ and getting stuck in overfitting, the proposed MoG-LASSO model adopts the L_1 norm regularization term, which can finish

feature selection simultaneously. The optimization problem of MoG-LASSO is:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (\sum_{k=1}^{K} \frac{\gamma_{ik}}{2\sigma_k^2})(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \|\boldsymbol{\beta}\|_1. \tag{15}$$

And one can add the column of 1s into the feature matrix X for the intercept term. From the above formulation, the model of MoG-LASSO can serve as weighted LASSO. Therefore, the warm starts strategy [38] can be used to tune the regularization parameter λ of MoG-LASSO, in which the optimal λ is contained in a decreasing sequence whose λ_{max} is the smallest value for zero regression estimations. Updated regression coefficients $\boldsymbol{\beta}$ will be returned to step E in each iteration.

In addition, the convergence of the parameter estimation sequence acquired by the EM algorithm and the corresponding sequence of the log-likelihood function have been proven in Refs. [39], [40], which can guarantee the effectiveness of the MoG-LASSO model. According to the proof in Ref. [40], the convergence value of the parameter estimation sequence from the EM algorithm is only the stable point of the log-likelihood function sequence. To solve this problem, a commonly used strategy that selects the one with the largest log-likelihood from different random initializations is applied in MoG-LASSO.

The MoG-LASSO will discriminate the noise of each Gaussian with different variances, while it is not the same as those regression feature selection methods whose loss functions are the absolute loss or the squared loss, etc. The weights of samples whose noises have large variances are set to small values according to Eq. (15). By the E step and M step, alternately, it estimates regression coefficients under good parameters of MoG and estimates parameters under good regression coefficients in each cycle. Its iterative process is always towards a better regression coefficient estimation. Additionally, after modeling unknown noises by the MoG distribution, the Bayesian maximum a posteriori estimation will be achieved with updated parameters of MoG. Therefore, MoG-LASSO can reduce the influence of unknown noise well and has good robustness.

4.3 Optimization of MoG-LASSO

The iterative method for solving the optimization problem named alternative search strategy (ASS) is used to solve MoG-LASSO. ASS divides variables into two disjoint blocks at first, and alternatively optimizes each of them under the other fixed. In each cycle of MoG-LASSO, the parameters of MoG are updated first under fixed $\boldsymbol{\beta}$. Then, $\boldsymbol{\beta}$ is updated under fixed parameters of MoG. The repeated process will end when a satisfactory result is obtained. $\boldsymbol{\beta}$ is updated according to the following optimization objective function:

$$\min_{\boldsymbol{\beta}} \frac{1}{2} \sum_{i=1}^{n} (\sum_{k=1}^{K} \frac{\gamma_{ik}}{2\sigma_k^2})(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|. \tag{16}$$

The coordinate descent method [11] is used to solve the above optimization problem, i.e.:

$$\begin{aligned} F &= \frac{1}{2} \sum_{i=1}^{n} (\sum_{k=1}^{K} \frac{\gamma_{ik}}{2\sigma_k^2})(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \\ &= \frac{1}{2} \sum_{i=1}^{n} (\sum_{k=1}^{K} \frac{\gamma_{ik}}{2\sigma_k^2})(y_i - \sum_{k' \neq j} x_{ik'}\beta_{k'} - x_{ij}\beta_j)^2 + \lambda \sum_{k' \neq j} |\beta_{k'}| + \\ &\quad \lambda |\beta_j|. \end{aligned}$$

Take the derivative of F with respect to β_j:

$$\begin{aligned} \frac{\partial F}{\partial \beta_j} =& \sum_{i=1}^{n} (\sum_{k=1}^{K} \frac{\gamma_{ik}}{2\sigma_k^2})(y_i - \sum_{k' \neq j} x_{ik'}\beta_{k'})(-x_{ij}) + \\ & \sum_{i=1}^{n} (\sum_{k=1}^{K} \frac{\gamma_{ik}}{2\sigma_k^2}) x_{ij}^2 \beta_j + \lambda \text{sign}(\beta_j). \end{aligned}$$

When $\frac{\partial F}{\partial \beta_j} = 0$, there is:

$$\sum_{i=1}^{n} (\sum_{k=1}^{K} \frac{\gamma_{ik}}{2\sigma_k^2}) x_{ij}^2 \beta_j = \sum_{i=1}^{n} (\sum_{k=1}^{K} \frac{\gamma_{ik}}{2\sigma_k^2})(y_i - \sum_{k' \neq j} x_{ik'}\beta_{k'}) x_{ij} - \lambda \text{sign}(\beta_j).$$

Let $z = \frac{\sum_{i=1}^{n} (\sum_{k=1}^{K} \frac{\gamma_{ik}}{2\sigma_k^2})(y_i - \sum_{k' \neq j} x_{ik'}\beta_{k'}) x_{ij}}{\sum_{i=1}^{n} (\sum_{k=1}^{K} \frac{\gamma_{ik}}{2\sigma_k^2}) x_{ij}^2}$, $g = \frac{\lambda}{\sum_{i=1}^{n} (\sum_{k=1}^{K} \frac{\gamma_{ik}}{2\sigma_k^2}) x_{ij}^2}$, and

$$\beta_j = \begin{cases} z - g & z > g \\ z + g & z < -g \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

where $\beta_j \in [0, z)$ or $(z, 0]$. When $z \neq 0$, β_j is related to λ. $\beta_j = 0$ when the value of λ is large. In the next loop, the parameters of MoG are updated according to Eq. (12) and Eq. (13).

4.4 MoG-LASSO Algorithm

In this section, the algorithm of MoG-LASSO is provided. First, MoG-LASSO calculates responsibilities γ_{ik} in the E step, and updates parameters of MoG in the M Step (Π and Σ). Second, it updates regression coefficients under updated weighted loss functions in the M Step ($\boldsymbol{\beta}$). Finally, the number of Gaussians K is updated. The MoG-LASSO algorithm iterates the E step and M step alternatively until the change in $\boldsymbol{\beta}$ between two consecutive iterations is smaller than a prespecified threshold ϵ or the maximum number of iterations ϱ is reached. Because the EM algorithm [39] can select any values as initial parameter values while estimating parameters of probability models containing the latent variable, MoG-LASSO randomly initializes parameters of MoG. The output of the algorithm is the final regression coefficient.

The main steps of the MoG-LASSO algorithm are summarized in Algorithm 1.

Algorithm 1: A Solving Algorithm for MoG-LASSO

Input: The training set $\mathbf{X} \in R^{n\times p}$ and $\mathbf{Y} \in R^n$, the parameter λ.

Output: The regression coefficient $\boldsymbol{\beta}$.

Step 1: Initialize: Randomly initialize Π, Σ, $\boldsymbol{\beta}$, τ, ϵ, ϱ, MoG number K. And $t = 1$.

Step 2.1: $t = t + 1$.

Step 2.2 (E Step): Update responsibilities γ_{ik} for $i = 1, 2, \cdots, n$; $k = 1, 2, \cdots, K$ by Eq. (11).

Step 2.3 (M Step (Π and Σ)): Update parameters of MoG Π and Σ by Eq. (12) and Eq. (13).

Step 2.4 (M Step ($\boldsymbol{\beta}$)): Update the regression coefficient $\boldsymbol{\beta}$ by Eq. (17) according to the updated parameters γ_{ik}, Π and Σ.

Step 2.5: Update the number of Gaussians K. If $\frac{|\sigma_i^2-\sigma_j^2|}{\sigma_i^2+\sigma_j^2} < \tau$ for some i and j, then the $i-$th and $j-$th Gaussian components are combined into a unique Gaussian according to Eq. (8) and Eq. (9). In addition, the $j-$th Gaussian parameters are then removed from Π and Σ. $K = K - 1$.

Step 3: end while

The running time of MoG-LASSO is mainly determined by step 2. In MoG-LASSO, the EM algorithm owns analytical expressions when estimating parameters of MoG, which will lead to lower computational complexity. Equations (11), (12) and (13) are calculated $nK + 2K$ times in each iteration of MoG-LASSO. In step 2.4, $O(n)$ calculations are made while updating the $j-$th component of the regression coefficient in the coordinate descent method, and the regression coefficient is the p-dimensional vector [11]. Then, the time complexity of computing $\{\beta_j\}_{j=1}^p$ once is $O(np)$. Assuming that $\{\beta_j\}_{j=1}^p$ is updated a^* times in step 2.4 and that there are b^* iterations in MoG-LASSO, the sum of the time complexities of the above steps is as follows: $O(b^*nK + 2b^*K + a^*b^*np)$. Hence, the time complexity of the MoG-LASSO method is $O(np)$.

5 Experiments

All experiments are conducted on the same PC with an Intel(R) Core(TM) i7-7700 3.60-GHz CPU, which has 64 cores and 16 GB of RAM, and run on Windows 10 and MATLAB (Version 2014a).

5.1 Experimental Results on Artificial Data and Benchmark Data Sets

5.1.1 Data Sets and Evaluation Indexes

To verify the validity of MoG-LASSO, 4 artificial data sets and 7 benchmark data sets are used [41], [42], [43]. MoG-LASSO is compared with several competitive regression estimation methods: LADE, LSE, MoG, MCCR, LAD-LASSO, LASSO, support vector machine (SVR), decision tree regressor (CART) and multilayer perceptron (MLP). For SVR, CART and MLP, the toolboxes in MATLAB are practical implementations, and their parameters are recommended in MATLAB. Specifically, the kernel function is linear and $C = 1$ in SVM. MLP uses different numbers of neurons depending on sample sizes, in which the number of neurons in the hidden layer is set as 10, 20 or 60.

Table 1. Artificial Data Sets

Artificial Data Sets	ε Distribution	♯Samples	β_h	Features	
				♯Relevant	♯Irrelevant
D1	20%-N(0,64) 30%-N(0,16) 50%-N(0,0.04)	200	β_1	4	16
D2	20%-N(0,64) 30%-U(-8,8) 50%-N(0,0.04)	500	β_1	4	16
D3	40%-N(0,64) 45%-U(-8,8) 15%-N(0,0.04)	500	β_1	4	16
D4	20%-N(0,64) 30%-U(-8,8) 50%-N(0,0.04)	50	β_2	4	96

Table 2. Benchmark Data Sets

Benchmark Data Sets	♯Samples	♯Features
Triazines	186	59
Pyrim	74	26
Eunite2001	367	15
Puma32h	8192	32
CBM-Compressor	11934	16
CBM-Turbine	11934	16
CT slices	100	384

For a given data set, it is generally impossible to know whether it contains unknown noise. Hence, we construct data sets that contain unknown noises according to Ref. [7]. The unknown noises are simulated by mixture noises that contain uniform noises and Gaussian noises. For artificial data sets, $\{\mathbf{x}_i\}_{i=1}^n$ are generated by $\mathrm{N_p}(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} = (\Sigma_{i^*j^*})$ with $\Sigma_{i^*j^*} = (\frac{1}{2})^{|i^*-j^*|}$. $y_i = \mathbf{x}_i^T\boldsymbol{\beta}_h + \varepsilon_i$, where $h = 1$ or 2, $\boldsymbol{\beta}_1 = (1, 2.5, 1.5, 2, 0, \cdots, 0)^T$ includes 16 irrelevant features, $\boldsymbol{\beta}_2 = (1, 2.5, 1.5, 2, 0, \cdots, 0)^T$ includes 96 irrelevant features, $\boldsymbol{\beta}_h$ is a real regression coefficient, and $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n)^T$ denotes the noise vector. Different artificial data sets are obtained according to the distribution of $\boldsymbol{\varepsilon}$, the content of noise whose variances are larger and the real regression coefficient. Table 1 shows 4 artificial data sets. Table 2 lists 7 benchmark data sets. Specifically, CBM-Compressor and CBM-Turbine come from the same benchmark data set, and they have different but remarkably similar response data. Moreover, the CT slices data set originally has 53,500 samples. To verify the validity of MoG-LASSO for small samples, 100 samples of original CT slices are selected as the training data.

For artificial data sets, the mean square error (MSE) that represents the difference between $\boldsymbol{\beta}^*$ and $\boldsymbol{\beta}_h$ is used to compare the robustness of algorithms,

$$MSE = \frac{1}{T}\sum_{t=1}^{T}\|\boldsymbol{\beta}_t^* - \boldsymbol{\beta}_h\|_2^2, \tag{18}$$

where T is repeated times and $\boldsymbol{\beta}_t^*$ is the regression coefficient acquired in the $t-$th experiment. Experiments are repeated 100 times in artificial data sets. If the MSE of a certain method is smaller, i.e., the difference between the estimated regression coefficients by this method and the real regression coefficient is smaller, then its robustness is better. At the same time, the mean number of selected irrelevant features is used to evaluate the sparsity of each method. When the value is closer to the true number of 0 in the real regression coefficient, the corresponding method is sparser. Moreover, CART and MLP are not considered in

artificial data sets because they cannot acquire the regression coefficient estimation.

The 10-fold cross-validation is applied on benchmark data sets to evaluate the performance of these methods. The most robust method is that having the lowest mean absolute error (MAE), the lowest root mean square error (RMSE), and the highest Willmott's index of agreement (WIA). They are in order of decreasing priority while evaluating robustness. These three indexes can represent the difference between predictions $\widehat{y_i}$ and observations y_i.

$$RMSE = \sqrt{\frac{1}{n'}\sum_{i=1}^{n'}(y_i - \widehat{y_i})^2}, \tag{19}$$

$$MAE = \frac{1}{n'}\sum_{i=1}^{n'}|y_i - \widehat{y_i}|, \tag{20}$$

$$WIA = 1 - \frac{\sum_{i=1}^{n'}(y_i - \widehat{y_i})^2}{\sum_{i=1}^{n'}(|y_i - \bar{y_i}| + |\widehat{y_i} - \bar{y_i}|)^2}, \tag{21}$$

where n' is the number of samples in the testing sets and $\bar{y_i}$ represents the mean of y_i in the testing sets. In addition, the best method in feature selection is the one having the highest mean number of selected irrelevant features and the most overlapped irrelevant features in 10-fold cross-validation.

The EM algorithm cannot guarantee that the global maxima is found [37]. Therefore, MoG-LASSO applies ten random initializations and selects the one that has the largest log-likelihood for alleviating this problem.

The comparative methods LADE, LSE, MoG, SVR, CART and MLP do not have the regularization parameter λ. And the regularization parameter of LAD-LASSO is set according to Ref. [20]. For LASSO and MCCR, their optimal parameters (λ for LASSO; λ and kernel scale parameter for MCCR) are learned via the routine 10-fold cross-validation under the squared error criterion. Due to the update of responsibilities and MoG parameters during the process of optimization, MoG-LASSO has different weights w_i in each iteration, which will lead to that different λ is selected from the 10-fold cross-validation under the squared error criterion for different iterations. And we regard the above λ as the optimal λ for each iteration. Hence, instead of a single optimal λ, MoG-LASSO obtains a set of optimal λ in current iterations, and its tuning of the optimal λ set will be twinborn with model solving.

5.1.2 Experimental Results on Artificial Data Sets

The feature selection ability and robustness of the proposed method are verified in this subsection.

(1) Feature Selection

First, the feature selection results of eight methods are compared and shown in Table 3, which is the mean number of selected irrelevant features.

In the D1, D2 and D3 data sets, 16 elements of the real regression coefficient are 0, i.e., these data sets have 16 irrelevant features. In D4, 96 elements of the real regression coefficient are 0, i.e., D4 has 96 irrelevant features. In experiments, because all coefficients of regression coefficient estimations are non-zero, LSE and MoG can not exclude irrelevant features in D1, D2 and D3, and LADE and SVR can not exclude irrelevant features in all data sets. Moreover, LSE and MoG are invalid in D4 because $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}^T\mathbf{W}\mathbf{X}$ are not invertible. Table 3 shows above experimental results by notations 0 and '−', respectively. It can be observed that the mean number of selected irrelevant features of LAD-LASSO is close to 0 in D1, D2 and D3 and close to 9 in D4. Comparing with MCCR, feature selection results of LASSO and MoG-LASSO are closer to 16 in D1, D2 and D3 and 96 in D4, and they generally exclude irrelevant features correctly. Hence, only MoG-LASSO and LASSO show good results in feature selection.

Table 3. Feature selection results on artificial data sets.

D1								
Models	SVR	LADE	LSE	MoG	MCCR	LAD-LASSO	LASSO	MoG-LASSO
Irrelevant Features Mean	0	0	0	0	8.26	0.51	12.61	10.07
D2								
Models	SVR	LADE	LSE	MoG	MCCR	LAD-LASSO	LASSO	MoG-LASSO
Irrelevant Features Mean	0	0	0	0	7.96	0.15	11.77	11.87
D3								
Models	SVR	LADE	LSE	MoG	MCCR	LAD-LASSO	LASSO	MoG-LASSO
Irrelevant Features Mean	0	0	0	0	8.82	0.09	11.83	10.02
D4								
Models	SVR	LADE	LSE	MoG	MCCR	LAD-LASSO	LASSO	MoG-LASSO
Irrelevant Features Mean	0	0	–	–	78.81	9.32	87.13	85.25

(2) Robustness Comparison

The robustness of eight methods is also compared. Fig. 1 shows the comparison results of MSE on artificial data sets, and its y-axis is on the log scale for convenient observation. LSE and MoG are invalid in D4, so corresponding experimental results are not shown. From Fig. 1, it can be seen that the MSE of MoG-LASSO is smallest on each data set, which means that the estimated regression coefficients of MoG-LASSO have little difference from the real regression coefficients. MoG, MCCR and LAD-LASSO achieve middling performance about MSE on all data sets. LSE obtains the largest MSE in D1, D2 and D3, and that of LADE is the largest in D4. In addition, the performance of SVR and LASSO are in the bottom three on most data sets. The experimental results support that the MSE of MoG-LASSO is smaller than that of its corresponding method, which cannot select irrelevant features from Fig. 1.

The D3 data set has more noises whose variances are larger than D2. From Fig. 1, it can be observed that the content of noise with large variances influences the experimental results. Compared with D2, the MSE of four methods increase dramatically in D3, e.g., it changes from the bottom of the x-axis to the top for LADE. MCCR, LAD-LASSO and LASSO have large MSE, and the MSE of MoG-LASSO, MCCR, LAD-LASSO and LASSO have little change.

Experimental results also show that the performance of MoG-LASSO does not decline dramatically as the content of noise with large variances increases. Hence, it can deal with unknown noise better and has good robustness. In addition, it can be shown that feature selection can improve robustness from experimental results.

The above experimental results on artificial data sets demonstrate that, regardless of the distribution of unknown

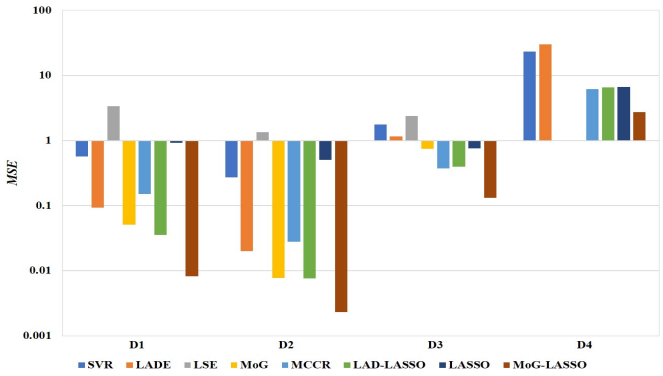


Fig. 1. Comparison of MSE on artificial data sets.

noise, the proposed MoG-LASSO has good robustness in comparison with the other seven methods. In addition, both MoG-LASSO and LASSO have good sparsity.

5.1.3 Experimental Results on Benchmark Data Sets

Because we do not know whether benchmark data sets contain noise or the noise distribution, we assume that they are clean. To verify the performance of the proposed model, the feature selection results and robustness are compared on original data sets first. Then, the experiments are performed on constructed noisy benchmark data sets.

(1) Results on original data sets

The experimental results of the ten models are shown in Table 4. The predicted results are the mean of ten folds. In Table 4, '$-$' denotes that the corresponding index is not available. LSE and MoG are invalid in six data sets because $\mathbf{X}^T\mathbf{X}$ and $\mathbf{X}^T\mathbf{W}\mathbf{X}$ are not invertible. In CBM-Compressor and CBM-Turbine, LADE and LAD-LASSO do not have experimental results in current experimental environments. MLP does not have the ability to select irrelevant features.

For feature selection, it can be seen that LSE and MoG could not obtain valid results in six data sets. In addition, they could not select irrelevant features in the remaining data set Puma32h. Neither LADE nor SVR finish feature selection in three data sets. For the remaining data sets, their the mean numbers of irrelevant features and overlapping irrelevant features are only around 3, even in CT slices with a large feature size. LAD-LASSO also only has the mean irrelevant feature number in four data sets, and its mean numbers of irrelevant features and overlapped irrelevant features are far below those of MCCR, LASSO and MoG-LASSO. In addition, it cannot acquire overlapping irrelevant features in Eunite2001. CART performs well in most benchmark data sets, but it selects 0 irrelevant features in Puma32h. Their feature selection ability is inferior to that of MCCR, LASSO and MoG-LASSO.

The mean number of irrelevant features of MoG-LASSO is larger than that of MCCR and LASSO in four data sets. It is smaller than that of MCCR in Puma32h and CBM and that of LASSO in Puma32h and CBM-Compressor. MoG-LASSO, LASSO and MCCR select many irrelevant features on most data sets, and they do not regard all features as irrelevant features. In addition, they have many overlapping irrelevant features, which means that their feature selection results have little difference between each fold. Therefore, on the original data sets, only the proposed MoG-LASSO, LASSO and MCCR have good performance on feature selection.

For the predicted results, LSE and MoG are only valid in Puma32h, and LADE and LAD-LASSO have results in five data sets. Especially for LADE, its MAE and RMSE are much larger than those of other models in Triazines. Other methods are valid on all data sets. From available experimental results, SVR has the worst performance in three data sets. CART and MLP achieve the best predicted results in three data sets, respectively, but their predicted results rank lower in some data sets. The performances of LASSO and MCCR are similar, and they behave not very good in terms of predicted results. MoG-LASSO takes the second place in Pyrim, Eunite2001 and CT slices, and ranks third in Triazines. It is at a moderate level on rest original benchmark data sets (Puma32h and CBM). Even so, MoG-LASSO does not have a distinct advantage in comparison with comparing methods on original data sets.

(2) Results on Noisy Data Sets

To verify the performance of the proposed method when unknown noises are embedded in data sets, $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n)^T$ is added to the training sets to construct noisy data sets. For each fold, 40% of the samples that are specified randomly are corrupted with Gaussian noise N(0,64), 30% are contaminated with uniformly distributed noise over [-10,10], and the remaining 30% are corrupted with Gaussian noise N(0,0.04). The content of noises whose variances are large is as high as 70%. The experimental results of the ten methods are shown in Table 5. The mean number difference is the absolute value of the difference in the mean number of irrelevant features between noisy and original data sets. The intersections between overlapping features of noisy and original data sets are represented by the intersection of overlapping features. The unavailable experimental results are also denoted by '$-$' in Table 5.

The original data sets are assumed to be clean, so their feature selection results can be regarded as baselines. We will verify the effectiveness of feature selection based on the mean number difference. If one has the smallest mean number difference, then unknown noises have the least influence on it. Similar to the original data sets, LSE and MoG are still only valid in Puma32h, and LADE and LAD-LASSO have results in five data sets. Moreover, CART and MLP cannot finish feature selection in one and seven data sets, respectively, and SVR remains poor performance on noisy data sets. Therefore, we mainly compare the difference in the mean number between MCCR, LASSO and MoG-LASSO, which are valid on all data sets.

From Table 5, it can be observed that the mean number difference of MCCR is smaller than that of MoG-LASSO in some data sets. But the number of overlapped irrelevant features in its intersections is far less than that of MoG-LASSO in four data sets. Comparing with LASSO, MoG-LASSO has the smaller mean number difference in all data sets. The mean number of irrelevant features of LASSO increases dramatically because it regards most features as irrelevant features, which leads to its intersections having many overlapping irrelevant features in some data sets. Hence, the feature results of MoG-LASSO are closer to those of original data sets.

For the predicted results, it can be seen that MoG-LASSO

Table 4. Experimental Results with the Fitted Parameter λ on Original Data Sets

Data sets	Models	Irrelevant features		MAE	RMSE	WIA
		Mean number	Overlapped irrelevant features			
	SVR	2.2	2(42,43)	0.0738	0.0983	0.9848
	CART	**53.1**	**48**(2,7,12-39,41-49,51-59)	**0.0267**	**0.0845**	0.9812
	MLP	–	–	0.1071	0.1940	0.9273
	LADE	2	2(42,43)	5.4389e+08	2.3075e+09	0.3218
Triazines	LSE/MoG	–	–	–	–	–
	MCCR	25.1	15(10,11,15,23,24,26-28,42,43,47,51,56,57,59)	0.0596	0.0892	0.9866
	LAD-LASSO	4.6	2(42,43)	0.0618	0.1459	0.9527
	LASSO	25.3	12(11,24,26-28,42,43,47,51,56,57,59)	0.0623	0.0886	0.9869
	MoG-LASSO	26.7	16(10,11,15,23,24,26-29,42,43,47,51,56,57,59)	0.0617	0.0886	**0.9870**
	SVR	1	1(24)	0.1068	0.1770	0.9455
	CART	**22.7**	**22**(2,4,6,7,9-26)	**0.0561**	0.1430	0.9542
	MLP	–	–	0.1878	0.2859	0.8352
	LADE	1	1(24)	0.1122	0.2098	0.9264
Pyrim	LSE/MoG	–	–	–	–	–
	MCCR	19.8	11(9,11-15,18,19,23,24,26)	0.0925	0.1592	0.9501
	LAD-LASSO	1.6	1(24)	0.1471	0.2310	0.8711
	LASSO	19.8	16(2,9-17,19,20,23-26)	0.0851	0.1409	0.9637
	MoG-LASSO	20.4	17(2,9-17,19,20,22-26)	0.0818	**0.1345**	**0.9656**
	SVR	2.3	2(7,9)	**0.0379**	0.0492	**0.9442**
	CART	**6**	**3**(3,7,9)	0.0704	0.1004	0.7407
	MLP	–	–	0.0423	0.0545	0.9303
	LADE	2	2(7,9)	0.0390	0.0507	0.9406
Eunite2001	LSE/MoG	–	–	–	–	–
	MCCR	2.4	2(7,9)	0.0381	**0.0491**	0.9435
	LAD-LASSO	1.9	0	0.0784	0.0994	0.5804
	LASSO	2.1	2(7,9)	0.0381	0.0492	0.9436
	MoG-LASSO	2.4	2(7,9)	0.0380	**0.0491**	0.9439
	SVR	0	0	0.1034	0.1262	0.2708
	CART	0	0	0.0083	0.0104	0.9693
	MLP	–	–	**0.0053**	**0.0066**	**0.9874**
	LADE	0	0	0.0209	0.0271	0.6642
Puma32h	LSE	0	0	0.0210	0.0268	0.6150
	MoG	0	0	0.0210	0.0268	0.6150
	MCCR	**28.4**	**23**(2-5,7-10,12-15,18,20-26,28,31,32)	0.0210	0.0267	0.6084
	LAD-LASSO	0	0	0.0209	0.0270	0.6653
	LASSO	27.5	21(2-5,7-10,12-15,18,20,21,23-26,28,31)	0.0210	0.0267	0.6076
	MoG-LASSO	27.4	19(2-5,7,8,10,12-15,20,21,23-26,28,31)	0.0210	0.0267	0.6079
	SVR	0	0	9.3827	12.6163	0.0154
	CART	4.1	**4**(2,7,9,12)	0.0012	0.0018	0.9963
	MLP	–	–	**4.6233e-05**	**6.6215e-05**	**1**
	LSE/MoG	–	–	–	–	–
CBM-Compressor	MCCR	2.6	2(9,12)	0.0065	0.0078	0.9006
	LADE/LAD-LASSO	–	–	–	–	–
	LASSO	**4.8**	**4**(2,3,9,12)	0.0063	0.0077	0.9016
	MoG-LASSO	2.1	2(9,12)	0.0064	0.0078	0.9029
	SVR	0	0	2.8624	3.4771	0.0033
	CART	4.5	**4**(2,7,9,12)	5.9625e-04	0.0012	0.9932
	MLP	–	–	**4.5090e-05**	**7.9584e-05**	**1**
	LSE/MoG	–	–	–	–	–
CBM-Turbine	MCCR	**8.1**	3(9,12,16)	0.0049	0.0059	0.6801
	LADE/LAD-LASSO	–	–	–	–	–
	LASSO	5.8	3(9,12,16)	0.0048	0.0057	0.7026
	MoG-LASSO	7.6	3(9,12,16)	0.0049	0.0058	0.6888
	SVR	3.7	3(255,303,375)	0.0970	**0.1189**	0.8792
	CART	**368.4**	**307**	**0.0935**	0.1643	0.7736
	MLP	–	–	0.1623	0.2128	0.7100
	LADE	3.3	3(255,303,375)	0.1019	0.1235	**0.8861**
CT slices	LSE/MoG	–	–	–	–	–
	MCCR	312.2	188	0.1063	0.1490	0.7962
	LAD-LASSO	148	1(87)	0.1972	0.2436	0.3492
	LASSO	314.3	200	0.1063	0.1464	0.8047
	MoG-LASSO	319.9	210	0.0970	0.1364	0.8105

Note: Due to the large feature size, only overlapping irrelevant feature numbers of some methods are given in CT slices.

has the best performance in terms of MAE, RMSE and WIA on all data sets. Its predicted results are much better than those of comparative methods. In addition, MoG-LASSO has the least differences in terms of MAE, RMSE and WIA compared with those on the original data sets.

The above experimental results on benchmark data sets with unknown noise demonstrate that the proposed MoG-LASSO has good feature selection ability and predicted performance. Therefore, it is robust in comparison with other methods.

5.2 Robust Face Recognition Based on Sparse Representation

5.2.1 MoG-LASSO on Face Recognition and Experiment Settings

In the context of face recognition, a grayscale face image can be represented by a vector given by stacking its columns. Assume that the training set contains n grayscale face images from C object classes and that the $c-$th object class has n_c images. The training set $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_C] \in R^{m\times n}$, where $\mathbf{X}_c = [\mathbf{v}_{c,1}, \mathbf{v}_{c,2}, \cdots, \mathbf{v}_{c,n_c}] \in R^{m\times n_c}$ and $n = \sum_{c=1}^{C} n_c$. Face recognition based on sparse representation first seeks the solution of the following problem:

$$\boldsymbol{\beta} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1.$$

Then, it classifies a test image $\mathbf{Y} \in R^m$ by

$$Label(\mathbf{Y}) = \arg\min_c r_c(\mathbf{Y}) = \arg\min_c \|\mathbf{Y} - \mathbf{X}\delta_c(\boldsymbol{\beta})\|_2,$$

where $\delta_c : R^n \to R^n$ is the characteristic function that selects the coefficients associated with the $c-$th class.

MoG-LASSO accomplishes face recognition under the frame of SRC. Hence, the optimization objective function is constructed as:

$$\boldsymbol{\beta} = \arg\min_{\boldsymbol{\beta}} \|\sqrt{\mathbf{W}} \odot (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1, \qquad (22)$$

Table 5. Experimental Results with the Fitted Parameter λ on Noisy Data Sets

Data sets	Models	Irrelevant features between noisy and original data sets		MAE	RMSE	WIA
		Mean number difference	Intersection of overlapped features			
Triazines	SVR	0.1	2(42,43)	0.9728	1.3920	0.3707
	CART	15.5	16(13,17,23-27,29,42,43,47,49,53,55,57,59)	3.4218	4.6362	0.0905
	MLP	–	–	2.6293	3.5033	0.2065
	LADE	0	2(42,43)	2.6670e+11	1.1607e+12	0.1368
	LSE/MoG	–	–	–	–	–
	MCCR	15.2	4(11,24,42,43)	0.4026	0.5599	0.6849
	LAD-LASSO	0.6	0	0.3695	0.4394	0.5369
	LASSO	30.6	9(11,24,27,42,43,51,56,57,59)	0.6351	0.7739	0.3092
	MoG-LASSO	25.4	12(10,23,24,27-29,42,43,51,56,57,59)	**0.2043**	**0.4282**	**0.8011**
Pyrim	SVR	0.1	1(24)	1.6629	2.1100	0.2605
	CART	8.7	2(20,24)	3.4605	4.1139	0.1588
	MLP	–	–	5.4177	6.8318	0.0735
	LADE	0	1(24)	4.2517	6.1774	0.1360
	LSE/MoG	–	–	–	–	–
	MCCR	1.6	2(23,24)	0.4347	0.5583	0.5762
	LAD-LASSO	0.1	0	0.5707	0.6573	0.3946
	LASSO	5.1	9(10,13,15,17,19,20,23-25)	0.5963	0.6968	0.3494
	MoG-LASSO	2.5	8(2,9,10,15,17,19,23,24)	**0.2692**	**0.3719**	**0.6714**
Eunite2001	SVR	0.1	2(7,9)	0.3081	0.3694	0.3349
	CART	0.9	2(7,9)	4.2117	5.2222	0.0304
	MLP	–	–	3.0333	4.3452	0.0387
	LADE	0	2(7,9)	0.3326	0.5133	0.3069
	LSE/MoG	–	–	–	–	–
	MCCR	5.6	2(7,9)	0.2486	0.3025	0.4479
	LAD-LASSO	0.5	0	0.1615	0.1854	0.3859
	LASSO	12.6	2(7,9)	0.2281	0.2752	0.3323
	MoG-LASSO	4.4	2(7,9)	**0.1002**	**0.1743**	**0.6717**
Puma32h	SVR	0	0	0.5327	0.6543	0.0765
	CART	0	0	4.2592	5.7450	0.0062
	MLP	–	–	1.1811	1.4787	0.0352
	LADE	0	0	0.0478	0.0596	0.4075
	LSE	0	0	0.2884	0.3603	0.0956
	MoG	0	0	0.0307	0.0388	0.5239
	MCCR	13.7	0	0.0336	0.0419	0.4540
	LAD-LASSO	0.1	0	0.0274	0.0346	**0.5367**
	LASSO	1.5	7(2,4,8,12,18,25,26)	0.0969	0.1138	0.2980
	MoG-LASSO	0.8	2(23,28)	**0.0231**	**0.0292**	0.4919
CBM-Compressor	SVR	0	0	448.6014	616.5788	3.5712e-04
	CART	0.6	4(2,7,9,12)	3.8133	5.0708	0.0039
	MLP	–	–	0.4011	0.5250	0.0432
	LSE/MoG	–	–	–	–	–
	MCCR	5.7	2(9,12)	0.0210	0.0265	0.5178
	LADE/LAD-LASSO	–	–	–	–	–
	LASSO	8.2	3(3,9,12)	0.0983	0.1222	0.2025
	MoG-LASSO	6.7	2(9,12)	**0.0203**	**0.0244**	**0.6619**
CBM-Turbine	SVR	0	0	80.4591	107.7606	2.7320e-04
	CART	0.5	4(2,7,9,12)	3.8262	5.0894	0.0019
	MLP	–	–	0.3956	0.5111	0.0213
	LSE/MoG	–	–	–	–	–
	MCCR	0.7	2(9,12)	0.0189	0.0231	0.3810
	LADE/LAD-LASSO	–	–	–	–	–
	LASSO	8.2	3(9,12,16)	0.0661	0.0783	0.2184
	MoG-LASSO	4.6	2(9,12)	**0.0096**	**0.0120**	**0.4418**
CT slices	SVR	1.4	3(255,303,375)	3.0146	3.6523	0.0974
	CART	5.8	215	4.6313	6.2319	0.0348
	MLP	–	–	6.5136	8.4299	0.0244
	LADE	0	3(255,303,375)	4.6011	5.4827	0.0356
	LSE/MoG	–	–	–	–	–
	MCCR	35.7	96	1.0141	1.3278	0.2649
	LAD-LASSO	1.6	0	0.4432	0.4866	0.3699
	LASSO	68.6	197	0.6199	0.8742	0.2845
	MoG-LASSO	58.6	197	**0.2218**	**0.2687**	**0.5287**

Note: Due to the large feature size, for some methods in CT slices, only overlapping irrelevant feature numbers of intersections are given.

where $\mathbf{W} = [\sum_{k=1}^{K} \frac{\gamma_{1k}}{2\sigma_k^2}, \sum_{k=1}^{K} \frac{\gamma_{2k}}{2\sigma_k^2}, \cdots, \sum_{k=1}^{K} \frac{\gamma_{mk}}{2\sigma_k^2}]^T$ is the weight vector. MoG-LASSO sets loss functions of pixels whose noises have larger variances smaller weights because their learning cost is larger and they may affect face recognition results.

Two available open databases, the Extended Yale B database [44] and the Feret database [45], are used in face recognition experiments. Table 6 summarizes above face databases from four attributes: the image size, number of classes and instances, and characteristics. The proposed MoG-LASSO is compared with state-of-the-art classifiers: LRC [35], NN [46], WSRC [47], CRC [48], SRC [33], Robust-SRC [33], SSRC [49], RLRC [50], CESR [51], RSC [52], SSEC [53], HQ_A [54], HQ_M [54] and NMR [55]. Among the above comparative methods, Ref. [55] claims that NMR achieves the best face recognition rates under extreme illumination conditions because the entire structural information and relationship of the error image are considered.

Table 6. Description of Face Databases

Databases	♯Classes	Size	♯Instances	Characteristics
Feret	40	112×92	240	posture, expression and illumination variations
Extended Yale B	38	192×168	2432	illumination variations

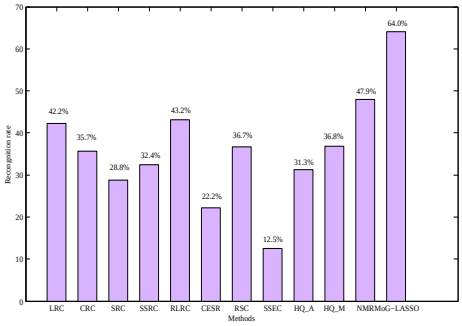
The parameter settings of the comparison methods follow the authors' suggestions. The regularization parameter for MoG-LASSO is chosen as $\lambda = 0.1$ for face recognition. The recognition rate is an evaluation index to verify the performance of each algorithm. The method whose recognition rate is larger has better robustness.

5.2.2 *Face Recognition with Illumination Changes*

First, face recognition rates are compared under extreme illumination conditions. We select subset 1 and subset 5 of Extended Yale B, which comes from Ref. [55] to conduct

experiments. Subset 1 with slight lighting conditions is used for training. Subset 5 with extreme illumination conditions is used for testing (see Fig. 2(a)). Except for the experimental result of MoG-LASSO, the experimental results of the comparative methods all come from the above reference. To be fair, the proposed method is also performed on the original face images, without any image preprocessing or feature extraction steps. The face recognition rates are shown in Fig. 2(b).

(a) Subset 5



(b) Recognition rates (%) of each classifier

Fig. 2. Recognition rates (%) under extreme illumination conditions on Subset 5 of the Extended Yale B database.

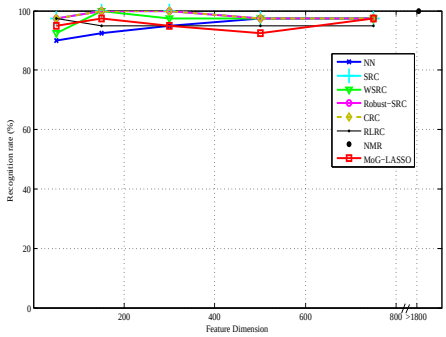
It can be observed that MoG-LASSO achieves the best recognition rate 64%. NMR performs the second best, and its recognition rate is only 47.9%. The third best recognition rate comes from RLRC (43.2%). Some robust sparse representation methods such as CESR, HQ_A, HQ_M, and SSEC which is exclusively designed for contiguous occlusion, are not very robust to extreme illumination changes. Differently and unexpectedly, the classical linear regression based method LRC and its robust version RLRC are more robust in the above situations. Therefore, MoG-LASSO performs well on face recognition and is more robust under extreme illumination changes in comparison with state-of-the-art face recognition methods.

The second experiment is conducted on the Feret database and the Extended Yale B database. For each subject of the Feret database, there is a fuzzy image that is badly affected by the illumination change, but the human eye can still recognize it. Fig. 3(a) is a typical fuzzy image. We choose fuzzy images of 40 individuals for testing and the remaining 200 images for training. Some images of the Extended Yale B database are in extreme lighting conditions, as shown in Fig. 3(b-d). They contain shadow noises, saturated noises or camera noises as high as 70%-90% and are hardly recognized by the human eye. Nine images affected slightly by illumination changes per subject for training and 18 images affected seriously per subject for testing were selected randomly. The three best methods (MoG-LASSO, NMR and RLRC) in the previous experiments and some commonly used face recognition methods, NN, SRC, WSRC, Robust-SRC and CRC, are compared.

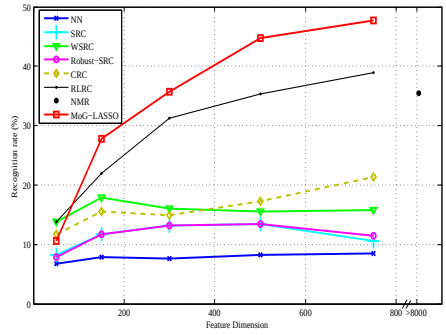
(a) (b) (c) (d)

Fig. 3. Typical fuzzy images of face databases.

For face recognition on sparse representation, downsampled images can perform as well as carefully engineered features [33]. Therefore, we adopt downsampled images and compare these methods on different feature space dimensions of 10×5, 15×10, 20×15, 25×20 and 30×25. NMR is still performed on the original face images because of the requirement of its model (so its result is only denoted as a single dot). The experimental results are shown in Fig. 4.



(a) The Feret database



(b) The Extended Yale B database

Fig. 4. Recognition rates (%) of different methods on different feature space dimensions.

On the Feret database, it can be observed from Fig. 4(a) that the face recognition results of the 8 methods are similar. All of them are close to 100%, i.e., they finish face recognition well when images contain few unknown noises.

From Fig. 4(b), it can be seen that the face recognition rate of each method increases as the dimension increases on the Extended Yale B database. It will be stable when the dimension is more than 500. It can also be observed that the recognition rates of MoG-LASSO outperform those of the six

methods from the 150 dimensional feature space. Its recognition rate changes between 35.67% and 44.74% for 300-500 dimensional feature spaces. These results are already superior to that of NMR, which is 35.4%. The maximum recognition rate of MoG-LASSO reaches 47.66% with the 750 dimensional feature space. The maximum recognition rates for NN, SRC, WSRC, Robust-SRC, CRC, RLRC and NMR are 8.48%, 13.45%, 17.84%, 13.45%, 21.35%, 38.89% and 35.4%, respectively. The best performance of MoG-LASSO consistently exceeds that of comparative methods.

All experimental results on face recognition databases demonstrate that the proposed MoG-LASSO has good robustness, even if images contain unknown noise as high as 70%-90%. In addition, MoG-LASSO has good face recognition results in comparison with state-of-the-art methods regardless of whether original face images are extracted features.

6 Conclusion

This paper proposes a regression feature selection method, MoG-LASSO, which can achieve feature selection and learner training simultaneously. It can model unknown noises well, and the loss function that corresponds to the noise distribution can be acquired. Additionally, feature selection can be achieved by adding the L_1 norm regularization term, and thus MoG-LASSO has better robustness and sparsity for data sets with irrelevant features. On open data sets, MoG-LASSO has better experimental results in comparison with common methods, even though the content of larger variance noises is as high as 70%-90%. Especially when face images are affected by illumination variations, MoG-LASSO still has a performance advantage over the state-of-the-art methods for face recognition. The experimental results support that MoG-LASSO has better performance and may provide a novel method for linear modeling.

However, the number of Gaussians is relative to unknown noises in data sets, and presently, there are no rules to determine how many Gaussians are needed. How to determine the number of Gaussians requires further investigation. In addition, MoG with a limited number of components has a limited ability to model unknown noise. Therefore, how to select a more adaptive distribution to model unknown noises is worth pursuing.

Acknowledgments

The authors would like to thank the editor and the anonymous reviewers for their critical and constructive comments and suggestions. In addition, the authors would like to thank Jian Yang and his team, who are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, for the provided database and source code of their paper. This work was supported by the National Natural Science Foundation of China (Nos. 61673249, U1805263), the Research Project Supported by Shanxi Scholarship Council of China (No. 2016-004), and the Graduate Education Innovation Project of Shanxi Province (No. 2020BY006).

References

[1] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, no. 6, pp. 1157–1182, 2003.
[2] Z. Zhou, *Machine Learning*. Beijing, China: Tsinghua University Press, 2016.
[3] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Berlin, Germany: Springer, 2009.
[4] C. Agostinelli, "Robust stepwise regression," *Journal of Applied Statistics*, vol. 29, no. 6, pp. 825–840, 2002.
[5] W. Yang, Y. Gao, Y. Shi, and L. Cao, "MRM-Lasso: a sparse multiview feature selection method via low-rank analysis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 11, pp. 2801–2815, 2015.
[6] J. Nakamura, *Image Sensors and Signal Processing for Digital Still Cameras*. Boca Raton, Florida, USA: CRC Press, 2005.
[7] D. Meng and F. D. L. Torre, "Robust matrix factorization with unknown noise," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1337–1344.
[8] A. N. Tikhonov and V. Y. Arsenin, *Solution of Ill-Posed Problems*. Washington, DC, USA: Winston and Sons, 1977.
[9] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, no. 1, pp. 267–288, 1996.
[10] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.
[11] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, "Pathwise coordinate optimization," *Annals of Applied Statistics*, vol. 1, no. 2, pp. 302–332, 2007.
[12] Z. Xu, H. Zhang, Y. Wang, X. Chang, and Y. Liang, "$L_{1/2}$ regularization," *Science China Information Sciences*, vol. 53, no. 6, pp. 1159–1169, 2010.
[13] J. Fan and H. Peng, "On nonconcave penalized likelihood with diverging number of parameters," *Annals of Statistics*, vol. 32, no. 3, pp. 928–961, 2004.
[14] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society, Series B*, vol. 67, no. 2, pp. 301–320, 2005.
[15] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society, Series B*, vol. 68, no. 1, pp. 49–67, 2006.
[16] D. Ming, C. Ding, and F. Nie, "A probabilistic derivation of LASSO and l12-norm feature selections," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 2019, pp. 4586–4593.
[17] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.
[18] C.-H. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *Annals of Statistics*, vol. 38, no. 2, pp. 894–942, 2010.
[19] Y. Feng, X. Huang, L. Shi, Y. Yang, and J. A. K. Suykens, "Learning with the maximum correntropy criterion induced losses for regression," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 993–1034, 2015.
[20] L. Wang, "L_1 penalized LAD estimator for high dimensional linear regression," *Journal of Multivariate Analysis*, vol. 120, no. 9, pp. 135–151, 2013.
[21] J. Xu and Z. Ying, "Simultaneous estimation and variable selection in median regression using Lasso-type penalty," *Annals of the Institute of Statistical Mathematics*, vol. 62, no. 3, pp. 487–514, 2010.
[22] Y. Li and J. Zhu, "L_1-norm quantile regression," *Journal of Computational and Graphical Statistics*, vol. 17, no. 1, pp. 163–185, 2008.
[23] A. Belloni and V. Chernozhukov, "L_1-penalized quantile regression in high-dimensional sparse models," *Annals of Statistics*, vol. 39, no. 1, pp. 82–130, 2011.
[24] J. Fan, Y. Fan, and E. Barut, "Adaptive robust variable selection," *Annals of Statistics*, vol. 42, no. 1, pp. 324–351, 2014.
[25] P. J. Huber, "Robust regression: asymptotics, conjectures and Monte Carlo," *Annals of Statistics*, vol. 1, no. 5, pp. 799–821, 1973.
[26] T. Yang, C. M. Gallagher, and C. S. McMahan, "A robust regression methodology via M-estimation," *Communications in Statistics-Theory and Methods*, vol. 48, no. 5, pp. 1092–1107, 2019.
[27] O. Arslan, "Weighted LAD-LASSO method for robust parameter estimation and variable selection in regression," *Computational Statistics and Data Analysis*, vol. 56, no. 6, pp. 1952–1965, 2012.

[28] A. Alfons, C. Croux, and S. Gelper, "Sparse least trimmed squares regression for analyzing high-dimensional large data sets," *Annals of Applied Statistics*, vol. 7, no. 1, pp. 226–248, 2013.
[29] T. M. Omara, "Weighted robust Lasso and adaptive elastic net method for regularization and variable selection in robust regression with optimal scaling transformations," *American Journal of Mathematics and Statistics*, vol. 7, no. 2, pp. 71–77, 2017.
[30] Y. Wang and L. Zhu, "Variable selection and parameter estimation via WLAD–SCAD with a diverging number of parameters," *Journal of the Korean Statistical Society*, vol. 46, no. 3, pp. 390–403, 2017.
[31] M. R. Osborne, B. Presnell, and B. A. Turlach, "A new approach to variable selection in least squares problems," *IMA Journal of Numerical Analysis*, vol. 20, no. 3, pp. 389–403, 2000.
[32] M. Ben-Ezra, S. Peleg, and M. Werman, "Real-time motion analysis with linear programming," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 32–52, 2000.
[33] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.
[34] J. Yang, L. Zhang, Y. Xu, and J.-y. Yang, "Beyond sparsity: the role of L1-optimizer in pattern classification," *Pattern Recognition*, vol. 45, no. 3, pp. 1104–1118, 2012.
[35] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 2106–2112, 2010.
[36] V. Maz'ya and G. Schmidt, "On approximate approximations using gaussian kernels," *IMA Journal of Numerical Analysis*, vol. 16, no. 1, pp. 13–29, 1996.
[37] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.
[38] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, 2010.
[39] H. Li, *Statistical Learning Method*. Beijing, China: Tsinghua University Press, 2012.
[40] R. M. Neal and G. E. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," *Learning in Graphical Models*, pp. 355–368, 1998.
[41] D. Dua and C. Graff, "UCI machine learning repository," http://archive.ics.uci.edu/ml, 2017, University of California, Irvine, School of Information and Computer Sciences.
[42] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2011, ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27.
[43] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 17, no. 2-3, pp. 255–287, 2011.
[44] K.-C. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.
[45] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The FERET database and evaluation procedure for face-recognition algorithms," *Image and Vision Computing*, vol. 16, no. 5, pp. 295–306, 1998.
[46] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
[47] C.-Y. Lu, H. Min, J. Gui, L. Zhu, and Y.-K. Lei, "Face recognition via weighted sparse representation," *Journal of Visual Communication and Image Representation*, vol. 24, no. 2, pp. 111–116, 2013.
[48] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation which helps face recognition?" in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 471–478.
[49] K. Jia, T.-H. Chan, and Y. Ma, "Robust and practical face recognition via structured sparsity," in *Proceedings of the European Conference on Computer Vision*, 2012, pp. 331–344.
[50] I. Naseem, R. Togneri, and M. Bennamoun, "Robust regression for face recognition," *Pattern Recognition*, vol. 45, no. 1, pp. 104–118, 2012.
[51] R. He, W.-S. Zheng, and B.-G. Hu, "Maximum correntropy criterion for robust face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1561–1576, 2011.
[52] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Robust sparse coding for face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 625–632.
[53] X.-X. Li, D.-Q. Dai, X.-F. Zhang, and C.-X. Ren, "Structured sparse error coding for face recognition with occlusion," *IEEE Transactions on Image Processing*, vol. 22, no. 5, pp. 1889–1999, 2013.
[54] R. He, W.-S. Zheng, T. Tan, and Z. Sun, "Half-quadratic based iterative minimization for robust sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 2, pp. 261–275, 2014.
[55] J. Yang, L. Luo, J. Qian, Y. Tai, F. Zhang, and Y. Xu, "Nuclear norm based matrix regression with applications to face recognition with occlusion and illumination changes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 1, pp. 156–171, 2017.

Yaqing Guo received the B.S. degree in computer science and technology from Shanxi University, China, in 2014. Now she is a Ph.D. candidate who major in computer science and technology from Shanxi University. Her current research interest includes machine learning.

Wenjian Wang received the BS degree in computer science from Shanxi University, China, in 1990, the MS degree in computer science from Hebei Polytechnic University, China, in 1993, and the PhD degree in applied mathematics from Xian Jiaotong University, China, in 2004. She is now a full-time professor and PhD supervisor of Shanxi University. She has been worked with the School of Computer and Information Technology at Shanxi University since 1993, where she was promoted as an associate professor in 2000 and as a full-time professor in 2004. She has published more than 150 academic papers. Her current research interests include machine learning, data mining and computational intelligence.

Xuejun Wang received the B.S. degree in mathematics and applied mathematics from Agricultural University of Hebei, Baoding, China, in 2011, and the M.S. degree in applied mathematics from China Jiliang University, Hangzhou, China, in 2014. He is currently pursuing the Ph.D. degree at the school of Computer and Information Technology, Shanxi University, Taiyuan, China. His current research interests include pattern recognition and image processing.