This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTSP.2018.2872460, IEEE Journal of Selected Topics in Signal Processing



# Robust Multinomial Logistic Regression Based on RPCA

Ming Yin, *Member, IEEE*, Deyu Zeng, Junbin Gao, Zongze Wu, Shengli Xie, *Senior Member, IEEE*

***Abstract*—Multiclass classification tasks are ubiquitous recently. In this scenario, the class label usually takes more than two possible discrete outcomes. As a simple and successful model, the multinomial logistic regression, also known as the softmax regression, is widely used in many multiclass classification applications. However, the existing method often experiences significant performance degradation when gross outliers are present in data features. To this end, in this paper, a novel robust multinomial logistic regression method is proposed by solving a rank minimization problem. In particular, the recovery of clean data and the logistic regression learning are conducted jointly. As such, the detection of the intra-sample outliers within data, by Robust Principal Component Analysis (RPCA), is performed in a *supervised* way. Although the problem is nonconvex and nonsmooth, the convergence is guaranteed by the recent theoretical advance of ADMM. Experimental analysis on synthetic and real-world data demonstrates that our method outperforms other state-of-the-art ones, in terms of classification accuracy.**

***Index Terms*—Robust classification methods, corrupted data, Logistic regression, missing data.**

## I. Introduction

In principle, logistic regression (LR) is a standard *probabilistic* statistical classification model, by modeling the binary class probability of a sample [7]. For multiclass problems, the logistic model is naturally extended to the multinomial logistic regression, also called softmax regression model[1].

For our purpose, we briefly review the definition of the softmax regression. Given sample $\mathbf{x}_i \in \mathbb{R}^{d_x}$, there are $K$ possible labels. Its label is usually encoded by a length-K one-hot vector, i.e., $\mathbf{y}_i = [0, 0, ..., 1, ..., 0]$, such that, if $\mathbf{x}_i$ belongs to the $k$-th class ($1 \leq k \leq K$), then only the $k$-th entry of $\mathbf{y}_i$ is one. In the softmax regression, the probability that $\mathbf{x}_i$ belongs to the $k$-th class is modelled by

$$p_k(\mathbf{x}_i|\theta_k) = \frac{\exp(\mathbf{x}_i^T \boldsymbol{\theta}_k)}{\sum_{j=1}^{K} \exp(\mathbf{x}_i^T \boldsymbol{\theta}_j)} \tag{1}$$

where $\boldsymbol{\theta}_k$ is the parameter vector associated with the $k$-th class. Accordingly, let $X = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N] \in \mathbb{R}^{d_x \times N}$ denote the samples matrix and $\Theta = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_K] \in \mathbb{R}^{d_x \times K}$ the parameter matrix. Given a set of training data $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, the softmax model will be utilized to learn the parameter matrix $\Theta$ by the maximum likelihood [2] [12] [13].

Ming Yin, Deyu Zeng, Zongze Wu and Shengli Xie are with School of Automation, Guangdong University of Technology, Guangzhou, 510006, China. E-mail: yiming@gdut.edu.cn, deyuzeng@hotmail.com, zzwu@gdut.edu.cn, shlxie@gdut.edu.cn. Junbin Gao is with The University of Sydney Business School, The University of Sydney, Camperdown, NSW 2006, Australia. E-mail: junbin.gao@sydney.edu.au. This work was supported by the National Natural Science Foundation of China through grants (Nos. 61703114, 61673126, 61876042 and 61773130), in part by Educational Commission of Guangdong Province, China (No.2017KTSCX059) and in part by the Science and Technology Plan Project of Guangdong Province (Nos. 2017A010101024 and 2015B010131014), China. Manuscript received xxx; revised xxx.

[1]http://ufldl.stanford.edu/wiki/index.php/Softmax

### A. Related Work

In fact, the realistic training data are inevitably corrupted by some noise that may be arbitrary, unbounded and even from the nonspecific distribution, named as "adversarial" [2] [6]. In that case, the maximal-likelihood estimate of logistic regression is very sensitive to the presence of outliers [20]. As a result, the performance of existing methods will degrade significantly when gross outliers are present in both labels and data features [20].

To address this drawback, the usual method is to identify samples which are influential for estimating parameters [1]. Specifically, in addressing outliers in labels, a complicated and more robust M-estimator [5] based loss function is proposed to learn the parameters rather than the standard loss function (negative log-likelihood) of logistic regression. Unfortunately, this method does not consider the scenarios where there exist outliers in data features and it is not suitable to the high-dimensional regime, in which the computational complexity is very commanding [6]. To remedy this issue, Feng *et al.* [6] proposed a robust logistic regression algorithm based on linear programming, termed as RoLR. Park and Konishi [18] presented a Mahalanobis distance based weighted log-likelihood function to enhance the robustness of logistic regression, which can reduce the influence of outliers within high-dimensional data efficiently. Furthermore, Xu *et al.* [24] proposed a novel logistic loss model, termed as robust bounded logistic regression, which is inspired by the Correntropy induced loss function (named *C-Loss*). Hung *et al.* [10] proposed a $\gamma$-logistic regression based on the minimum $\gamma$-divergence estimation. By allowing the data itself to select the trimming value, a self-selecting robust logistic regression model was proposed in work [11]. In addition, to improve the out-of-sample performance of the classifier, a distributionally robust logistic regression model was formulated to minimize a worst-case expected logloss function [21]. Recently, in order to address the issue that the noisy input data are used in training model, Huang *et al.* [9] presented a novel linear regression with an effective convex formulation that used recent advances on the rank minimization [25].

[2]Generally speaking, there are two types of errors in training data. One is related to input $X$, and the other refers to labelling errors. In this paper, we focus the former.

Although the existing logistic regression methods are claimed to achieve considerable performance for multiclass tasks, they have one vital limitation. In the presence of gross errors in input data, and/or for the case of missing data, the existing approaches are not effective to deal with yet.

### B. Our Work

To circumvent the limitations of the existing methods and inspired by the recent advance of the rank minimization, a novel robust softmax regression method is proposed to handle the errors in inputs instead of fitting or pre-screening. Our main contributions can be briefly summarized into three folds.

- Different from the traditional errors-in-variables models [8] which assume input noises are independent data-by-data, our method recognizes the global non-Gaussian errors in terms of low-rank property.
- By virtue of the low-rank constraint, the proposed approach exploits Robust Principal Component Analysis (RPCA) [23] to detect the intra-sample outliers within training data in a *supervised* way.
- We extend the proposed approach to be able to deal with missing data in regression, wherein some entries of inputs are unknown.

The remainder of the paper is organized as follows. Section II presents a novel robust multinomial logistic regression model and its optimization, convergence and complexity analysis. Simulation experiments are conducted on the synthetic data sets and several real data sets in Section III. Finally, the conclusions are given in Section IV.

### C. Notations

In this paper, we use the uppercase letters, e.g., $u, v, ...$ to represent the scalars, and boldface uppercase letters, e.g., $\mathbf{u}, \mathbf{v}, ...$ to represent vectors. We use capital letters (such as $U$) to represent a matrix. $\mathbf{u}_j$ denotes the $j$-th column of a matrix $U$. The transpose and inverse of matrix $U$ are expressed as $U^T$ and $U^{-1}$, respectively. $U_{ij}$ represents the entry at the $i$-th row and $j$-th column of the matrix $U$. $|U_{ij}|$ represents an absolute value of an element $U_{ij}$. $||U||_0$ denotes the $\ell_0$ norm of $U$, $||U||_1$ denotes the $\ell_1$ norm of $U$ defined by the sum of the absolute values of the elements in $U$, $\|U\|_F$ denotes the Frobenius norm, $\|U\|_*$ denotes the trace norm or nuclear norm[3], and $\langle U, V\rangle$ represents the inner product of two matrices. In addition, the meaning of special letters will be explained in the paper.

## II. Robust Multinomial Logistic Regression

In this section, we first introduce the objective function of our robust softmax regression (RSR) and then present an optimization method for RSR.

[3]The nuclear norm of a matrix is the sum of the singular values of the matrix

### A. The proposed method

For the standard multinomial logistic regression, the parameter $\Theta$ is often used to study the relation between $X$ and the label $Y = \{y_{ik}\} \in \mathbb{R}^{N\times K}$, which can be learned by minimizing the so-called log likelihood (cross-entropy) function as follows,

$$L(X, \Theta) = -\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik} \log p_k(\mathbf{x}_i|\boldsymbol{\theta}_k), \tag{2}$$

where $y_{ik}$ is the entry of the $i$-th sample associated with the $k$-th class in one-hot coding.

By the advantage of RPCA [23], we here assume that the corrupted input data $X = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N] \in \mathbb{R}^{d_x\times N}$ follows the noise model $X = D + E$, where $D$ is a matrix containing the clean information and $E$ denotes the interference information. According to the RPCA theory, $D$ and $E$ are seen to be low-rank clean data components and sparse outliers whose magnitudes arbitrarily large, respectively. Past research, see [16], has shown that the algorithm of learning $\Theta$ cannot be robust enough when $X$ is corrupted by $E$ that is not clearly known. To this end, we proposed RSR to enhance the robustness of the multinomial logistic regression by solving the following problem.

$$\begin{aligned}\min_{\Theta,D,E}\ & L(D, \Theta) + \beta * \text{rank}(D) + \lambda\|E\|_0,\\ &\text{s.t. } X = D + E,\end{aligned} \tag{3}$$

where the rank constraint and $\ell_0$-norm are used to model the low-rank component and sparse outlier component of data respectively. $\beta$ and $\lambda$ are tradeoff parameters to balance the effects of all terms. Intuitively, it will be better to apply the clean data $D$ to train a model for its parameters. Generally, the noise-free components $D$ or/and outliers are unknown, thus the existing methods often utilize $X$ to learn $\Theta$. However, it may lead to a biased estimation of $\Theta$ due to the presence of outliers. In contrast, our RSR can efficiently handle this issue by explicitly factorizing $X$ into the sum of noise-free components and outliers. As such, only the clean data $D$ is utilized to learn $\Theta$.

***Remark:*** Note that our method is distinct from cleaning the data by RPCA and then applying the logistic regression to the noise-free data, because RPCA cleaning process is done in a unsupervised manner, while ours is conducted in a *supervised* way. In other words, the clean data $D$ can preserve the information of $X$ that is maximally correlated with $Y$. As such, the outlier component $E$ is able to effectively model the errors inside the data $X$, which is not directly related to $Y$ information. Under this scenario, both the recovered training data and testing data are robust to noise or corruptions, and it favors to discriminate the samples from different classes.

It is not easy to optimize problem (3) since the rank and $\ell_0$-norm are both discontinuous and non-convex. Then the corresponding convex surrogates are adopted, and the problem can be re-formulated as follows.

$$\min_{\Theta,D,E}\ L(D, \Theta) + \beta\|D\|_* + \lambda\|E\|_1,\ \text{s.t. } X = D + E, \tag{4}$$

where the $\|D\|_*$ is the nuclear norm of D and $\|E\|_1$ is $\ell_1$-norm.

**Algorithm 1:** Algorithm for solving Eq. 5.

**Input** $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$. **Initialization:** $D_0 = X,\ E_0 = X - D_0,\ \Theta_0 = 0,\ \mu = 0.1,\ \mu_{\max} = 1e9,\ \rho = 1.01;\ \varepsilon_1 = 10^{-4}$, $\varepsilon_2 = 10^{-5}$.

**While** not converged $(t = 0, 1, ...)$ **do**

1) Update $\Theta$ according to $\Theta := \Theta - \alpha\nabla_\Theta J_\Theta$ with 10 steps;
2) Update $D$ according to $D := D - \gamma\nabla_D J_D$ with 10 steps;
3) Update $Z$ by $Z^* = U_z\Delta_{\frac{\beta}{\mu}}(\Sigma_z)V_z^T$, where $U_z\Sigma_z V_z^T$ is the SVD of $W_2 + \mu D$ and $\Delta_\tau(\cdot)$ is the SVT operator [3] defined by $\Delta_\tau(\Sigma) = \text{diag}(\text{sgn}(\Sigma_{ii})(|\Sigma_{ii}| - \tau))$.
4) Update $E$ by $E^* = \mathcal{S}_{\lambda/\mu}(X - D + W_1/\mu)$, where $\mathcal{S}_\tau(\cdot)$ is the shrinkage operator [14] defined by $\mathcal{S}_\tau(E_{ij}) = \text{sgn}(E_{ij})\max\{|E_{ij}| - \tau, 0\}$.
5) Update $W_1, W_2$ and $\mu$ as follows;

$$W_1 \leftarrow W_1 + \mu(X - D - E), W_2 \leftarrow W_2 + \mu(Z - D), \mu \leftarrow \min(\rho\mu, \mu_{\max}).$$

6) Check convergence: If $\|X - D_t - E_t\| / \|X\| < \varepsilon_1$ and $\max\left\{ \|E_{t+1} - E_t\|_\infty / \|X\|_\infty,\ \|Z_{t+1} - Z_t\|_\infty / \|X\|_\infty,\ \|\Theta_{t+1} - \Theta_t\|_\infty / \|X\|_\infty \right\} \le \varepsilon_2$, then break.

**End while**

**Output** $\Theta^*, D^*, Z^*, E^*$

### B. Optimization of RSR

To separate $D$ from two terms, we introduce an auxiliary variable $Z$ and let $Z = D$. We will use the ADMM (Alternating Direction Method of Multipliers) method to optimize this problem. Then, the augmented Lagrangian function of the proposed problem (4) is given by,

$$\begin{aligned}\min_{\mathbf{\Theta},D,Z,E} \mathcal{L}_\mu = \min_{\mathbf{\Theta},D,Z,E}\ & L(D,\Theta) + \beta\|Z\|_* + \lambda\|E\|_1 \\ & + \langle W_1, X - D - E\rangle + \langle W_2, Z - D\rangle \\ & + \frac{\mu}{2}(\|X - D - E\|_F^2 + \|Z - D\|_F^2), \end{aligned} \quad (5)$$

where $W_1$ and $W_2$ are the Lagrange multiplier matrices with compatible dimension and $\mu$ is the penalty parameter. The overall algorithm is summarized in Algorithm 1 and its detailed optimization is elaborated in Appendix. Empirically our algorithm has strong convergence behavior indeed, shown in Figure 2(a).

*Theorem 1 (Convergence of Algorithm 1): The algorithm 1 will converge globally for any sufficiently large $\mu$. That is, starting from any $y^0, Z^0, E^0$, it generates a sequence that is bounded, having at least one limit point, and that each limit point $(y^*, Z^*, E^*)$ is stationary point of $\mathcal{L}_{\mu^*}$, namely, $0 \in \partial\mathcal{L}_{\mu^*}(y^*, Z^*, E^*)$ [22].*

The proof of Theorem 1 is provided in Appendix.

### C. Algorithm Complexity Analysis

In order to assess the complexity of the proposed algorithm, we not only give the complexity of the proposed algorithm represented by $\mathcal{O}(\cdot)$ but also list each iteration of the algorithm operation conditions, as shown in Table I [4]. The total computational complexity of the algorithm is $\mathcal{O}(\min(d_x NRt, d_x Nt^2))$.

[4] $R$ is is the lowest rank for $W_2^{(k)} + \mu D^{(k)}$ and $t$ denotes the number of iterations.

### D. Classification by RSR

By Algorithm 1, the classifier with $\Theta \in \mathbb{R}^{d_x \times K}$ can be learned, even when gross outliers are present. Given a test sample $\mathbf{x}_i \in \mathbb{R}^{d_x}$ as an input, the probability that the sample belongs to $k$-th class can be calculated by Eq. (1). The result $p \in \mathbb{R}^K$ represents the probability that the data belongs to each class. The index of the maximum value of $p$ is the classification prediction.

### E. RSR for Missing Data

To handle the problem that input data $X$ with missing elements, especially for the case that some elements of $X$ are unknown, our method can be readily extended. We here refer to it as "RSR-Missing", which solves the following problem.

$$\begin{aligned}\min_{\mathbf{\Theta},D,E}\ & L(D,\Theta) + \beta\|D\|_* + \lambda\|E\|_1, \\ \text{s.t. } & \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(D + E), \end{aligned} \quad (6)$$

where $\Omega$ represents the set of coordinates $(i, j)$ of all non-missing elements. $\mathcal{P}_\Omega(X)$ represents a projection operator that sets the elements of matrix which are out of $\Omega$ to be zero and contains the elements in $\Omega$ of matrix. The augmented Lagrange function of the above problem is formulated as following.

$$\begin{aligned}\arg\min_{\mathbf{\Theta},D,Z,E}\ & L(D,\Theta) + \beta\|Z\|_* + \lambda\|E\|_1 \\ & + \langle W_1, \mathcal{P}_\Omega(X - D - E)\rangle + \langle W_2, Z - D\rangle \\ & + \frac{\mu}{2}(\|\mathcal{P}_\Omega(X - D - E)\|_F^2 + \|Z - D\|_F^2). \end{aligned} \quad (7)$$

By slightly modifying Algorithm 1, we can easily optimize problem (7) with the ADMM as well. And the convergence of algorithm can also be guaranteed.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTSP.2018.2872460, IEEE Journal of Selected Topics in Signal Processing



TABLE I: Computational complexity analysis for single iteration of the algorithm

| | Addition | Multiplication | Complexity |
|---|---|---|---|
| Compute $\Theta$ | $Nt + d_x t$ | $d_x Nt + Nt + d_x t$ | $\mathcal{O}(d_x Nt)$ |
| Compute D | $Nt + d_x N$ | $d_x Nt + Nt$ | $\mathcal{O}(d_x Nt)$ |
| Compute Z | $R^2$ | $d_x N^2 + d_x NR + R^2$ | $\mathcal{O}(d_x NR)$ |
| Compute E | $d_x N$ | $d_x N$ | $\mathcal{O}(d_x N)$ |

### F. The connection to Robust Regression [9]

It is noteworthy that although the proposed model is related to the work [9], i.e., robust regression, they are distinct. The commonality between both approaches is that the low-rank and sparse properties are used to improve the robustness of the algorithms. On other hand, the main differences lie in two aspects. First, the former focuses a categorical output rather than a continuous one. Second, our model equipped with logistic regression actually runs the result through a special non-linear function. From this viewpoint, the proposed model is significantly distinct from the work [9].

In particular, from the objective functions of both approaches, we can see these two methods are different, especially for the fidelity terms. In essence, linear regression learns the parameter in a least square metric, assuming the error obeys a Gaussian distribution. In contrast, the conditional distribution in logistic regression is a Bernoulli distribution rather than a Gaussian distribution, since the dependent variable is binary. Moreover, the predicted values by logistic regression are probabilities and are thus restricted to (0,1) through the logistic distribution function. Furthermore, logistic regression is another generalized linear model that adopts the non-linear function (Sigmoid function or Softmax function) to fit discrete values [17][5]. As shown in Fig. 1, the linear regression(Fig. 1 (a)-(c)) cannot fit the discrete value well, in contrast, the shape of the sigmoid function is better from Fig. 1 (d). In addition, to highlight the differences, the extensive experiments are conducted successively.

## III. Experimental Results

In this section, several experiments are performed to test our RSR method to verify its effectiveness. In particular, to evaluate more comprehensively the performance of RSR, in Section III-A, we firstly present the multi-class classification results on a synthetic data and then human faces in Section III-B, i.e., Extended YaleB dataset. In Section III-C and Section III-D, the classification results of object recognition and hand-written digits are shown. In addition, we present the performance of the RSR-missing algorithm in Section III-E. Four state-of-the-art approaches of multi-class classification are selected to compare with ours. K-nearest neighbors (K-NN) is well known as a non-parametric method for classification and regression. Its output is achieved by using the nearest $k$ samples. RPCA+Softmax aims to first perform RPCA [4] on the input data, and then learn from the clean data using multinomial logistic regression (or Softmax regression). For convenience, we call Robust Regression [9] as RR, while ours as RSR.

[5]The sigmoid function is a special case of the logistic function. See p.96-97 of [17]

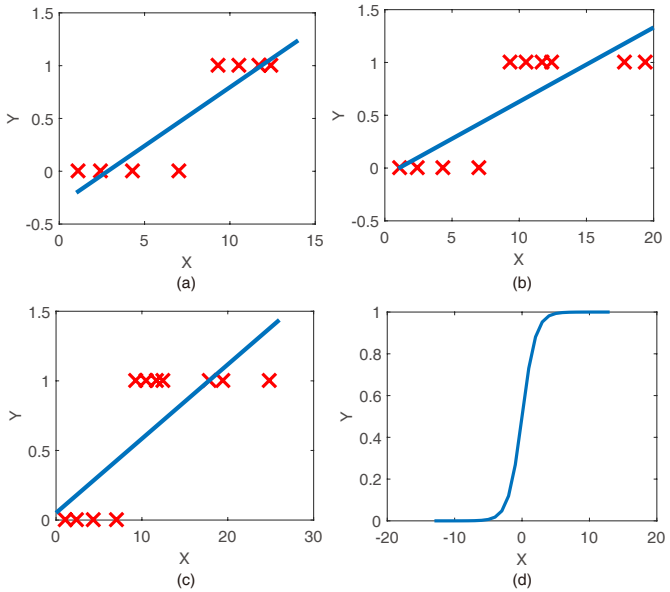



Fig. 1: (a)-(c) show the lines which fit the discrete data points by Least Square Regression. (d) shows the shape of sigmoid function.

### A. Experiment on Synthetic Data

*1) Regression for Synthetic Data:* In this part, the recovery capabilities of RSR algorithm is first validated with the training data of different sizes and varying degrees of interference. We randomly generate $D_0 \in \mathbb{R}^{m\times m}$ and $\Theta_0 \in \mathbb{R}^{m\times 5}$. $E_0 \in \mathbb{R}^{m\times m}$ denotes the interference that is sparse with percent 5% and 10% of non-zero values, respectively. Then, the synthetic data $X$ is defined by $X = D_0 + E_0$. For $i$-th data point, we simulate its outcome by Eq. (1). The Relative Absolute Error (RAE) is used to evaluate the result of regression in our experiment. From Table II, it is observed that RSR achieves better performance in terms of two metrics against RR. Best results are in bold.

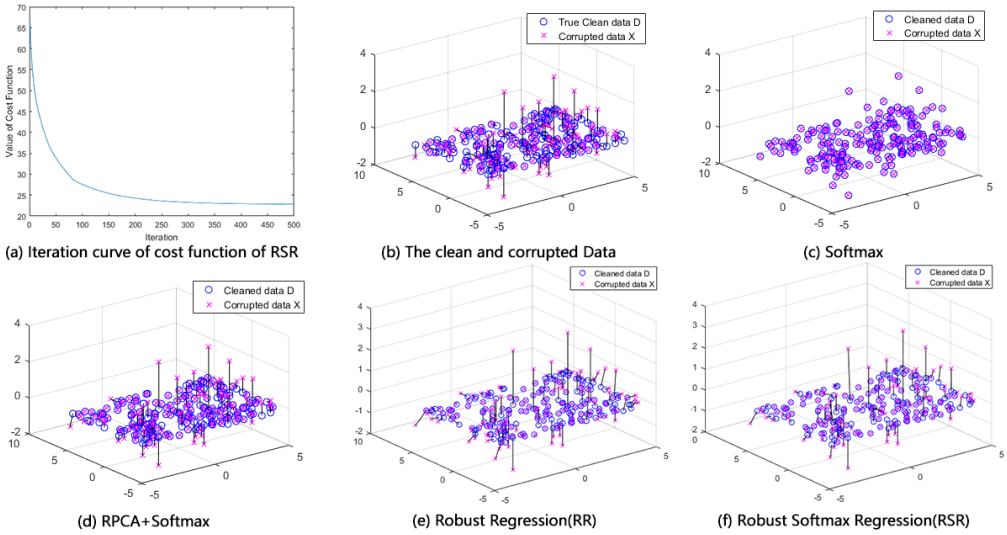
*2) Classification for Synthetic Data:* Next, we synthesized 200 three-dimensional samples $D \in \mathbb{R}^{3\times 200}$ where the first two components were generated from a uniform distribution between [0,6], and the entries at the third row were all 0. $\Theta \in \mathbb{R}^{4\times 5}$ was randomly generated and used as the classifier. The error term, $E \in \mathbb{R}^{3\times 200}$, was generated as follows. For 50 random sample, we added random Gaussian noise ($\sim N(0,1)$) in the second dimension, which simulates the intra noise. Similarly, for another 50 random samples, the random Gaussian noise ($\sim N(0,1)$) is added to the third dimension. This simulates noise outside the variables [9]. As for the label of the synthetic data, it is determined by the maximal value

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTSP.2018.2872460, IEEE Journal of Selected Topics in Signal Processing



TABLE II: Results on synthetic data in regression experiment.

| m | rank($D_0$) | $\|E_0\|_0$ | $RAE_D$ of RSR | $RAE_\Theta$ of RSR | $RAE_D$ of RR | $RAE_\Theta$ of RR |
|---|---|---|---|---|---|---|
| 100 | 5 | 500 | **3.03** $\times 10^{-3}$ | **0.969** | 0.642 | 1.990 |
| 200 | 10 | 2000 | **3.45** $\times 10^{-2}$ | **0.980** | 0.959 | 1.094 |
| 100 | 5 | 1000 | **2.41** $\times 10^{-12}$ | **0.984** | 0.940 | 1.153 |
| 200 | 10 | 4000 | **0.483** | **0.984** | 1.317 | 1.177 |




(a) Iteration curve of cost function of RSR (b) The clean and corrupted Data (c) Softmax

(d) RPCA+Softmax (e) Robust Regression(RR) (f) Robust Softmax Regression(RSR)

Fig. 2: (a) show the Iteration curve of RSR. (b) Original 3D synthetic dataset. (c)-(f) show the data cleaned by algorithms.

TABLE III: Relative Absolute Error (RAE) of $D$ and $\Theta$, and recognition rate (%) of synthetic data (10 trials).

| | K-NN | Softmax | RPCA+Softmax | RR | RSR |
|---|---|---|---|---|---|
| ACC(%) | $91.17 \pm 2.56$ | $91.56 \pm 1.88$ | $92.89 \pm 1.99$ | $84.33 \pm 2.19$ | $\mathbf{94.5 \pm 1.54}$ |

of $A = \exp([\mathbf{1}^T; D]^T \Theta) \in \mathbb{R}^{200\times 5}$ [6]. In Matlab notation, the label matrix $Y$ was generated by $[\sim, Y] = \max(A)$, i.e., turning $Y$ into a 0-1 matrix. For example, if the $i$-th case is in class 1, we have the five-dimensional 0-1 vector $\mathbf{y}_i = (1, 0, 0, 0, 0)$. Fig. 2(b) shows the clean data $D$ with blue "o", while the corrupted data $X$ with purple "×". To show it clearly, the black line is used to link $D$ with $X$ as they are the same sample before and after corruption respectively. As for the convergence of our algorithm, the curve of iteration vs. the objective cost of RSR is given in Fig. 2(a). By this, it can be seen the algorithm achieves convergence quickly. In addition, Fig. 2(c)-(f) visualize the results of the recovery of $D$ from $X$, by the different methods. As well, the data cleaned by these methods are marked by blue "o" too. As can be seen from Fig. 2, our algorithm is able to better clean up the data with interference. The recognition results are summarized in Table III, where the scores are calculated by averaging the values after 10 trials. Specifically, the results are reported in a mean $\pm$ deviation way. Obviously, it shows that our algorithm attains the higher accuracy against others. More importantly, our algorithm does better than the RR [9] thanks to its non-linear property.

[6]The $\exp(\cdot)$ denotes an exponential function, whose value of a matrix is defined by the exponential function of each element in the matrix.

### B. Experiment on Face Recognition

*1) YaleB dataset:* Next, we test our approach on the Extended YaleB dataset, which contains 2414 different face images with corruption by the shadow, belonging to 38 classes. The size of each image is normalized to $32 \times 32$ by down-sampling. First, in order to test the recovery capability of different methods, the artificial interferences are added to the data, which include 30% random pixel crops and $8 \times 8$ random block corruptions, respectively. In Fig. 3(a), the recovery of clean data $D$ and $E$ from the original data are given by different methods. It is observed that each algorithm has its capability to remove the outlier in data. From this viewpoint, the RR approach removed some details of the data. Fig. 3(b) shows the recovery of $D$ and $E$ from YaleB with 30% random pixel cropping. As can be seen, $D$ achieved by RSR is a good remedy for the pixel interference, while the RPCA and RR method cannot completely remove the interference within images. This owes to that our $D$ is attained in a supervised learning manner, and thus $D$ will complement each other with the same class of information.

Next, as shown in Fig. 3(c), the recoveries of $D$ and $E$ from YaleB dataset corrupted by $8 \times 8$ black blocks are shown. Although RR and RPCA can also get rid of the partial black blocks, there are still some obvious blocks remaining in $D$. Compared to RR and RPCA, our RSR removes the black blocks efficiently and achieves better visual effects.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTSP.2018.2872460, IEEE Journal of Selected Topics in Signal Processing



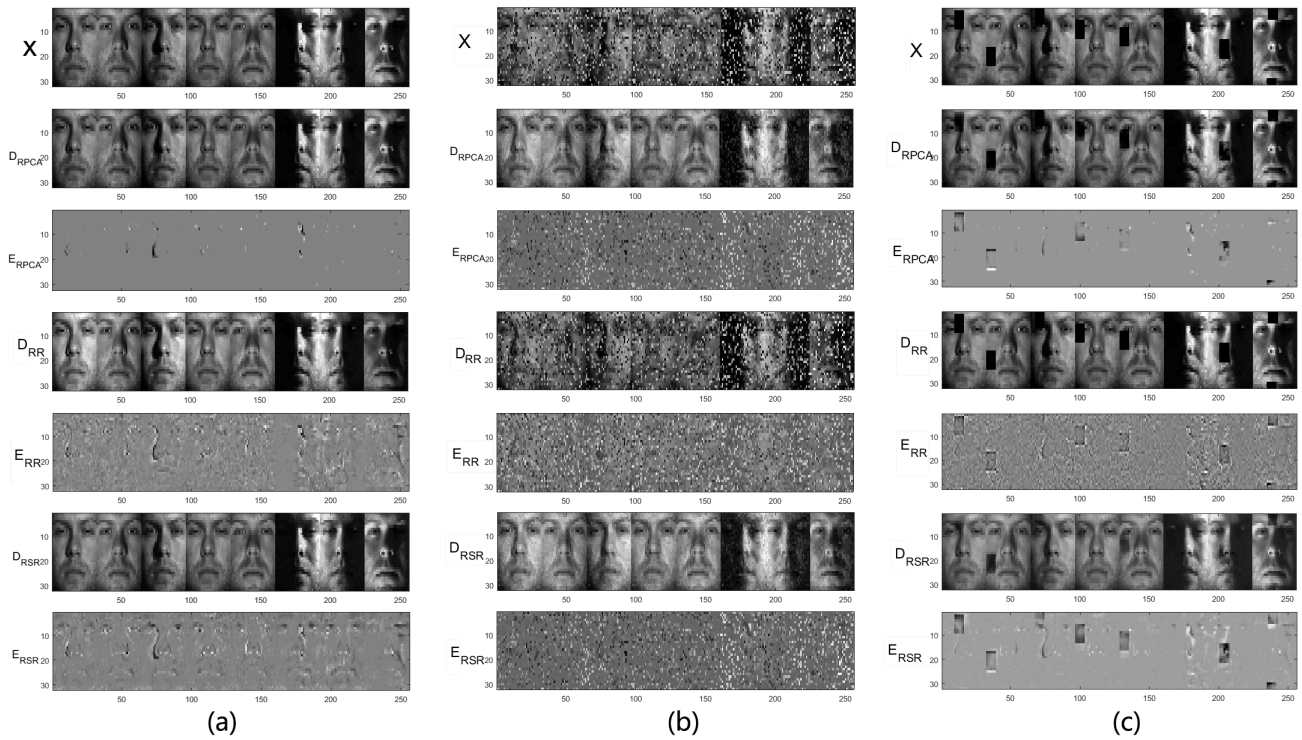



Fig. 3: Experiments on the recovery of YaleB dataset.(a) the results on YaleB dataset by different methods. (b) the results on YaleB dataset with 30% random pixel crop. (c) the results on YaleB datasets with $8 \times 8$ pixel corruptions.

Furthermore, we focus on the face recognition on YaleB. By randomly selecting some part classes of the YaleB dataset, we build four different test datasets, i.e., 5-sub, 10-sub, 20-sub and 38-sub. The half of data are for training and the rest for testing. This setting are kept for the subsequent classification experiments. Then the classification accuracy and training time are both reported in Table IV. It is observed that RSR outperforms others in accuracy score of all tests. As for the computational time, it is reasonable to see that the training time becomes large along with the increasing of the number of data. While for our method, the cost of training time is far smaller than that of RPCA+Softmax.

TABLE IV: The recognition rate (%) and training time on YaleB.

| | K-NN | Softmax | RPCA+Softmax | RR | RSR |
|---|---|---|---|---|---|
| 5-sub | | | | | |
| ACC | 88.75 | 93.75 | 93.13 | 92.50 | **98.75** |
| Time(s) | 0.0064 | 9.46 | 15.80 | 8.82 | 15.40 |
| 10-sub | | | | | |
| ACC | 83.44 | 90.94 | 90.63 | 92.19 | **96.56** |
| Time(s) | 0.0075 | 47.69 | 71.28 | 12.88 | 38.42 |
| 20-sub | | | | | |
| ACC | 74.22 | 92.56 | 92.34 | 93.75 | **96.25** |
| Time(s) | 0.011 | 338.31 | 570.02 | 26.71 | 162.87 |
| 38-sub | | | | | |
| ACC | 72.41 | 87.90 | 88.07 | 90.81 | **94.20** |
| Time(s) | 0.027 | 1396.2 | 2117.2 | 89.04 | 484.38 |

Furthermore, some artificial interferences are added to the YaleB dataset to simulate non-Gaussian noise. First, we discard some pixels of the image randomly, i.e., random pixel cropping. As such, a blurry image is obtained. In the experiments, we use the data corrupted to train the classifier, and then test on the data without/with interference, respectively. For clarity, we denote $ACC_o$ the accuracy on the data without corruption, while $ACC_c$ the accuracy on the disturbed data. Fig. 4 shows the results of the classification, where the classifiers are learned with data corrupted by different ratios of interference. In particular, the Fig. 4 (b) shows the classification results on the data with corruptions, which are the same as that of training data. It is observed that the RSR algorithm achieves the highest score not only for the data with interference but also for the data without interference.

TABLE V: The recognition rate (%) and training time on YaleB with $8 \times 8$ block corruptions.

| | K-NN | Softmax | RPCA+Softmax | RR | RSR |
|---|---|---|---|---|---|
| $ACC_c$ | 59.15 | 82.68 | 83.76 | 90.72 | **92.38** |
| $ACC_o$ | 72.99 | 89.72 | 89.72 | 93.04 | **95.44** |
| Time | 0.1147s | 1165.4s | 1545.7s | 90.20s | 450.85s |

Finally, we test on YaleB corrupted by $8 \times 8$ black blocks. The results are given in Table V. From the Table, we can see our algorithm outperforms against others consistently. As for the training time, the cost of our approach is relatively reasonable, which is much smaller than that of Softmax and RPCA+Softmax.

### C. Experiment on Object Recognition

The COIL20, from Columbia Object Image Library, contains 20 grayscale of different objects. In order to avoid the influences of background, each object is placed on a black background with an electric turntable desktop for shooting.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTSP.2018.2872460, IEEE Journal of Selected Topics in Signal Processing



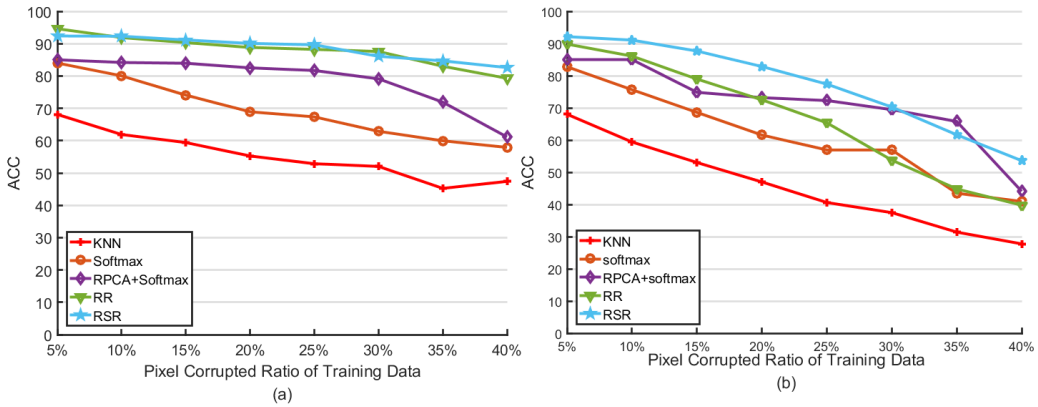



Fig. 4: (a) Results on the test data without interference by different methods. (b) Results on the test data with random pixel crop by different methods.

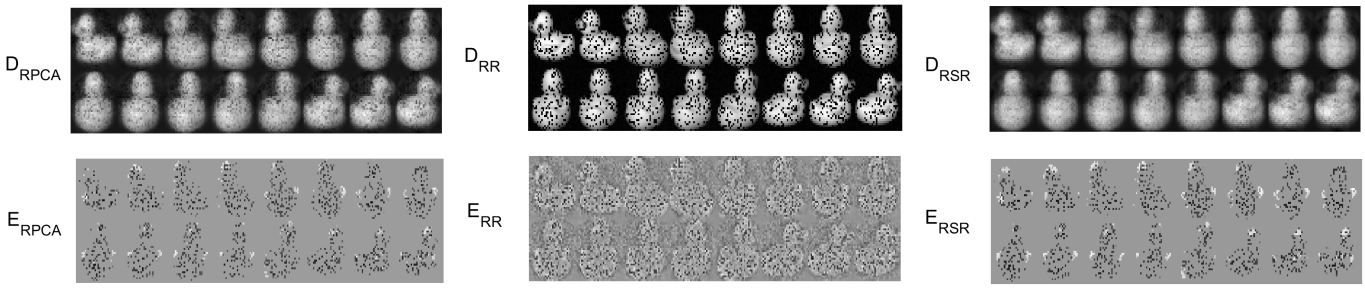



Fig. 5: The recovery of COIL20 with 20% random pixel cropping by RPCA, RR and RSR respectively.

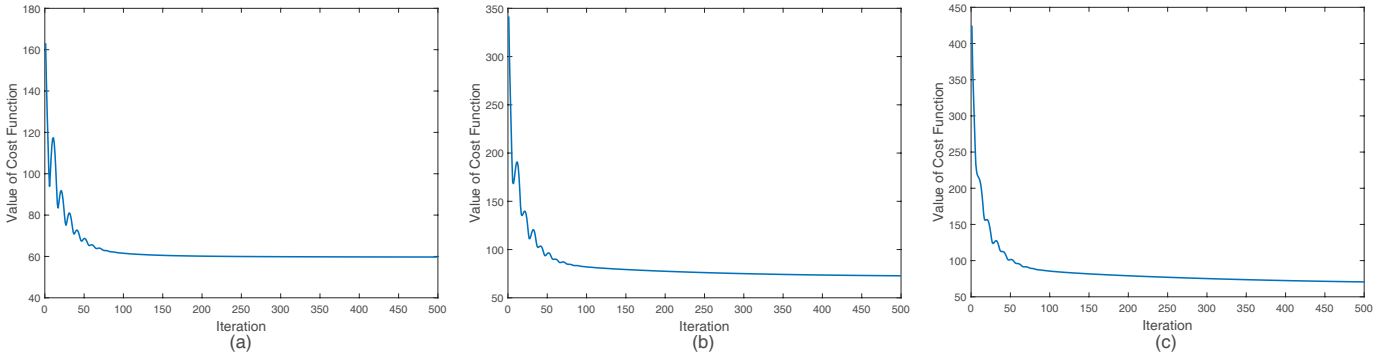



Fig. 6: The iteration curve of RSR on COIL20 dataset with 0%, 20% and 40% pixel corruption, respectively.

The electric turntable can rotated 360 degrees. Every 5 degrees of rotation of the turntable, a fixed position camera captures an image. 72 images were taken for each object. In this dataset, there are 20 different objects and 1440 images in total. To effectively recover the clean data, several methods are applied to COIL20 with 20% random pixel cropping. The $D$ and $E$ learned are also shown in Fig. 5. Furthermore, we conducted classification tests on COIL20 with different ratios of random pixel cropping. The results are shown in Table VI, where our algorithm achieves a higher gain over the standard softmax, in terms of classification accuracy. The best scores are marked in bold. Compared with other methods, RSR shows a promising robustness, consistently producing a good results. To show the convergence of our algorithm, the curves of iteration vs. objective value are provided in Fig. 6. From the figure, it can be seen that the algorithm converges fast.

### *D. Experiment on Handwriting Recognition*

The MNIST database (Modified National Institute of Standards and Technology database), shown in Fig. 7, is a large database of handwritten digits that is commonly used for evaluating various classifiers. Here a subset of MNIST that contains 4000 samples is utilized in the experiments. Similar to COIL20, different ratios of random pixel cropping are added to MNIST. Then, the classification tests are conducted by the methods. The results of recognition rate are given in Table VII. It shows that RSR algorithm achieves the highest accuracy in all cases.

TABLE VI: The recognition rate on COIL20 with different ratios of random pixel cropping. The percentages in the brackets denotes the portion. Bigger value (%) indicates better performance.

| pixel corruption(%) | K-NN | Softmax | RPCA+Softmax | RR | RSR |
|---|---|---|---|---|---|
| 0 | 98.75 | 98.89 | 97.50 | 97.78 | **99.44** |
| 5 | 98.61 | 98.61 | 98.89 | 94.31 | **99.72** |
| 10 | 98.75 | 97.78 | 98.75 | 98.47 | **98.89** |
| 15 | 97.50 | 97.36 | 97.36 | 89.31 | **97.64** |
| 20 | 96.53 | 95.12 | **97.36** | 85.64 | 97.08 |
| 25 | 96.94 | 95.14 | **97.36** | 85.97 | 97.22 |
| 30 | 95.97 | 93.33 | 95.28 | 86.81 | **96.81** |
| 35 | 93.42 | 92.50 | 94.58 | 85.69 | **94.58** |
| 40 | 91.67 | 91.25 | 86.39 | 85.56 | **93.33** |

Fig. 7: Samples of MNIST dataset.

### E. Experiment on Missing Dataset

In order to verify the performance of our RSR-missing algorithm on missing data, we here conduct the following experiments on COIL20 and PIE respectively. For simulating the case of missing, different ratios of pixels are cropped randomly from each sample and the corresponding pixels are set to be zero as the missing part of the sample. For MC+Softmax, we first apply matrix completion [15] to missing data and then use Softmax to classify the recovered data.

*1) COIL20 Dataset:* Here, this experiment randomly removes 10% to 70% pixels in each sample of COIL20. The classification results are shown in Table VIII, and our method still achieves the highest accuracy.

TABLE VII: The recognition rate on MNIST with different ratios of random pixel cropping. The percentages in the brackets denotes the portion. Bigger value (%) indicates better performance.

| pixel corruption(%) | K-NN | Softmax | RPCA+ Softmax | RR | RSR |
|---|---|---|---|---|---|
| 0 | 85.50 | 80.75 | 80.80 | 79.10 | **85.85** |
| 5 | 84.50 | 79.60 | 80.65 | 78.20 | **85.05** |
| 10 | 83.65 | 80.15 | 79.80 | 78.80 | **84.35** |
| 15 | 82.75 | 78.15 | 79.40 | 76.85 | **83.00** |
| 20 | 80.35 | 78.35 | 76.00 | 76.80 | **82.55** |
| 25 | 79.75 | 74.45 | 74.00 | 75.00 | **80.35** |
| 30 | 79.70 | 73.10 | 66.65 | 74.85 | **79.00** |
| 35 | 77.15 | 71.65 | 53.35 | 73.85 | **79.45** |
| 40 | 76.80 | 71.25 | 34.80 | 72.00 | **77.25** |

TABLE VIII: The recognition rate on COIL20 with different ratios of missing. The percentages in the brackets denotes the portion. Bigger value (%) indicates better performance.

| pixel missing (%) | Softmax | MC+Softmax | RR-missing | RSR-missing |
|---|---|---|---|---|
| 10 | 97.78 | 97.64 | 93.06 | **99.31** |
| 20 | 96.53 | 97.08 | 89.17 | **97.50** |
| 30 | 94.58 | 96.53 | 87.92 | **96.81** |
| 40 | 90.83 | 92.50 | 85.27 | **95.41** |
| 50 | 86.94 | 89.31 | 81.11 | **91.39** |
| 60 | 79.61 | 82.70 | 77.08 | **85.69** |
| 70 | 75.00 | 70.28 | 72.36 | **79.58** |

*2) PIE Dataset:* The PIE dataset, created by Carnegie Mellon University in the United States, contains 41,368 multi-colored, light and facial expressions of 68 volunteers. The changes in attitude and lighting are also captured under strictly controlled conditions. It has gradually become an important test set in the field of face recognition. This experiment adopts a subset of this dataset containing 490 samples as training and 505 samples for test. Fig. 8 shows the sample of original PIE. The pixels of training data are discarded randomly from 10% to 30%. By our approach, the recovery of $D$ from training data are shown in Figs. 9- 11, respectively. It is observed that our algorithm is able to complete the missing pixels effectively. Then, the classification results by different algorithms are given in Fig. 12 too. Similarly, our method obtains the best performance over other methods.

Fig. 8: Samples of PIE dataset.

### F. Parameter Setting

As usual, grid search is applied for all methods in our experiments to find the suitable parameter. For K-NN method, the parameter $K$ is set from 1 to 10 to find the best result. For RPCA, we choose the key parameter $\lambda$ as suggested by [23], i.e., $m^{\frac{1}{2}}$ ($m$ is the maximum dimension of the input matrix). For other compared methods, the most suitable parameters are searched exhaustedly in the range of $[10^{-6}, 10^{-5}, ..., 10^{6}]$.

### G. Parameter sensitivity

In the proposed model, there are two parameters, $\beta$ and $\lambda$, trading off the logistic regression term, low-rank term and sparsity term. In this section, we test the effect of these parameters on the performance of our model, by using MNIST with random pixel corruption. Specifically, the parameters are set to be different values, e.g., $[10^{-6}, 10^{-5}, ..., 10^{5}, 10^{6}]$. Fig. 13 shows how the accuracy of RSR changes with different values of $\beta$ and $\lambda$, while keeping others fixed. It can be seen from the figure that our algorithm performs consistently better except when $\beta$ is much larger than $\lambda$. In other words, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSTSP.2018.2872460, IEEE Journal of Selected Topics in Signal Processing



Fig. 9: PIE with 10% pixels missing and data recovered by the RSRmissing.

Fig. 10: PIE with 20% pixels missing and data recovered by the RSRmissing.

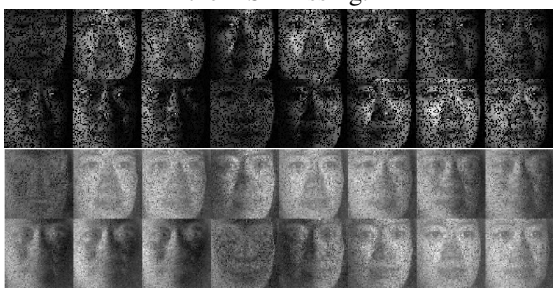


Fig. 11: PIE with 30% pixels missing and data recovered by the RSRmissing.

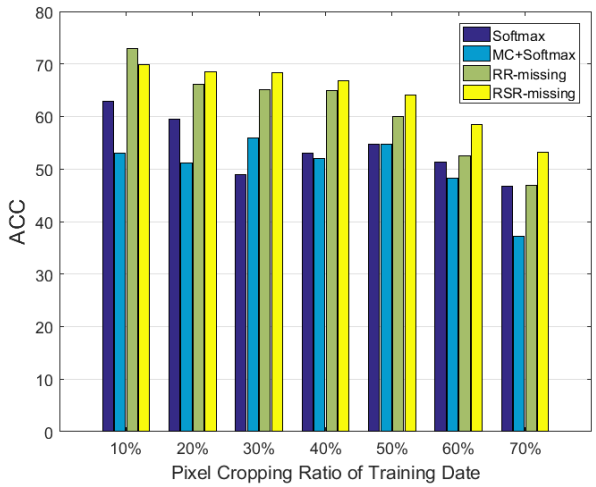
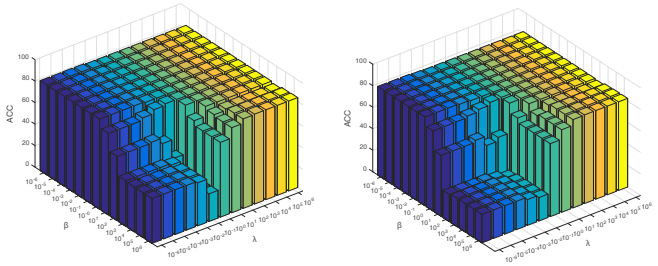



Fig. 12: Results of classification on PIE with different ratio of missing.



(a) (b)

Fig. 13: The accuracy of RSR with varying parameters, on MNIST dataset with random pixel corruption 10% and 20%, respectively.

proposed method is pretty stable regardless of the choice of its parameters $\beta$ and $\lambda$, as long as the parameters are selected in an proper range.

## IV. Conclusion

In this paper, a novel robust multinomial logistic regression method is proposed to handle the data corrupted by arbitrary outliers, such as non-Gaussian noise. Specifically, by solving a rank minimization problem, we jointly learn a logistic regression for multiple classes while eliminating the outliers that are not related to labels or regression outputs. Powered by RPCA, the proposed method is capable of recovering the clean data from the corrupted one and learning the regression parameters efficiently as well. Furthermore, we extend our method to missing data, namely, RSR-missing, to tackle the missing elements in the input data. Experimental results shows that our method outperforms other state-of-the-art ones in several multiple classes classification tasks.

## Appendix

### A. Optimization of RSR

The optimization algorithm is elaborated as follows.

- Updating $\Theta$
  The sub-problem for $\Theta$ is thus formulated by,

$$J_\Theta = \arg\min_{\Theta} L(D, \Theta)$$
$$= \arg\min_{\Theta} -\sum_{i=1}^{N}\sum_{k=1}^{K} Y_{ik} \log p_k(\mathbf{d}_i|\theta_k), \quad (8)$$

  where $D = [\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_N]$.
  To solve the above problem, as usual, we resort to an iterative optimization algorithm such as gradient descent or L-BFGS. By taking derivative of $J_\Theta$ w.r.t. $\Theta$, one can have $\nabla_\Theta J_\Theta = \frac{1}{N} D(P-Y)$. Armed with this formula for the derivative, one can then plug it into an algorithm such as gradient descent, and finally minimize the $J_\Theta$. That is, with the standard implementation of gradient descent, in each iteration, we would perform the update by $\Theta := \Theta - \alpha\nabla_\Theta J_\Theta$, where $\alpha$ denotes the learning rate of $\Theta$. We repeat ten step update for the sub-problem. This will be used as the exact solution for $\Theta$ with the current data $D$.
- Updating $D$

The corresponding sub-problem can be written as,

$$\begin{aligned} J_D = &\arg\min_D \ L(D,\Theta) \\ &+ \langle W_1, X - D - E\rangle + \langle W_2, Z - D\rangle \\ &+ \frac{\mu}{2}(\|X - D - E\|_F^2 + \|Z - D\|_F^2). \end{aligned} \tag{9}$$

The derivative of this function w.r.t. $D$ is given as follows.

$$\begin{aligned} \nabla_D J_D = &\frac{1}{N}\Theta(P - Y)^T \\ &+ \mu(2D + E - Z - X) - W_1 - W_2. \end{aligned} \tag{10}$$

Similarly, the $D$ is updated by, $D := D - \gamma\nabla_D J_D$. Only ten step in updating $D$ is performed, similar to $\Theta$.

- Updating $Z$
The sub-problem of $Z$ is formulated as follows.

$$\arg\min_Z \beta\|Z\|_* + \langle W_2, Z - D\rangle + \frac{\mu}{2}\|Z - D\|_F^2. \tag{11}$$

This problem has a closed-form solution, illustrated as,

$$Z^* = U_z \Delta_{\frac{\beta}{\mu}}(\Sigma_z) V_z^T, \tag{12}$$

where $U_z\Sigma_z V_z^T$ is the SVD of $W_2 + \mu D$ and $\Delta_\tau(\cdot)$ is the SVT operator [3] defined by,

$$\Delta_\tau(\Sigma) = \text{diag}(\text{sgn}(\Sigma_{ii})(|\Sigma_{ii}| - \tau)).$$

- Updating $E$
The sub-problem of $E$ is updated as follows.

$$\begin{aligned} \arg\min_E &\lambda\|E\|_1 + \langle W_1, X - D - E\rangle \\ &+ \frac{\mu}{2}\|X - D - E\|_F^2 \\ &= \mathcal{S}_{\lambda/\mu}(X - D + W_1/\mu). \end{aligned} \tag{13}$$

where $\mathcal{S}_\tau(\cdot)$ is the shrinkage operator [14] defined by $\mathcal{S}_\tau(E_{ij}) = \text{sgn}(E_{ij})\max\{|E_{ij}| - \tau, 0\}$.

- Updating $W_1, W_2$ and $\mu$

$$\begin{aligned} W_1 &\leftarrow W_1 + \mu(X - D - E), \\ W_2 &\leftarrow W_2 + \mu(Z - D), \\ \mu &\leftarrow \min(\rho\mu, \mu_{\max}). \end{aligned} \tag{14}$$

where $\rho > 1$ is a constant and $\mu_{\max}$ is the upper bound of $\mu$.

### B. Detailed Proof of Theorem 1

*(Convergence of Algorithm 1) The algorithm 1 will converge globally for any sufficiently large $\mu$. That is, starting from any $y^0, Z^0, E^0$, it generates a sequence that is bounded, has at least one limit point, and that each limit point $(y^*, Z^*, E^*)$ is stationary point of $\mathcal{L}_{\mu^*}$, namely, $0 \in \partial\mathcal{L}_{\mu^*}(y^*, Z^*, E^*)$ [22].*

*Proof:* Our model is as follows.

$$\begin{aligned} &\min_{\mathbf{\Theta}, D, Z, E} \ L(D,\Theta) + \beta\|Z\|_* + \lambda\|E\|_1, \\ &\text{s.t. } X = D + E, Z = D. \end{aligned} \tag{15}$$

Rewrite the optimization to a standard form, we have

$$\begin{aligned} &\min_{y,Z,E} \ h(y) + \beta f_1(Z) + \lambda f_2(E), \\ &\text{s.t. } \begin{bmatrix}\mathbf{I}\\ \mathbf{I}\end{bmatrix} D + \begin{bmatrix}-\mathbf{I} & 0\\ 0 & \mathbf{I}\end{bmatrix}\begin{bmatrix}Z\\ E\end{bmatrix} = \begin{bmatrix}0\\ X\end{bmatrix}, \end{aligned} \tag{16}$$

where $y \to (D, \Theta)$, $f_1(\cdot) = \|\cdot\|_*$ ,$f_2(\cdot) = \|\cdot\|_1$, and the $\mathbf{I}$ denotes the identity matrix with suitable dimension.

Denote by $\mathbf{A} = \begin{bmatrix}-\mathbf{I} & 0\\ 0 & \mathbf{I}\end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix}\mathbf{I}\\ \mathbf{I}\end{bmatrix}$.

Model (16) satisfies the following five conditions. •

1) The objective function is coercive as the coercivity of $\|\cdot\|_*$ and $\|\cdot\|_1$. The feasible set is
$$\mathcal{F} := \left\{(D, Z, E) \in \mathbb{R}^{d_x \times N} : \mathbf{B}D + \mathbf{A}\begin{bmatrix}Z\\ E\end{bmatrix} = \begin{bmatrix}0\\ X\end{bmatrix}\right\}.$$
If the set is bounded, the the coercivity of the objective is trivial. If it is unbounded, then when $(D, Z, E) \in \mathcal{F}$ and $\|D, Z, E\| \to \infty$, due to the coercivity of of $\|\cdot\|_1$, the objective function $h(y) + \beta f_1(Z) + \lambda f_2(E)$ is coercive over this set;
2) The objective function is feasible as $\text{Im}(\mathbf{A}) \subseteq \text{Im}(\mathbf{B})$ where $\text{Im}(\cdot)$ returns the image of a matrix;
3) $h(y)$ is Lipschitz differentiable with constants $L_h$ because of the smoothness of the softmax function.
4) Because $\mathbf{A}$ and $\mathbf{B}$ both have full column ranks, the objective function meets Lipschitz sub-minimization paths.
5) As we know $f_2(E) = \|E\|_1$ is a continuous and piecewise linear function, and $f_1(Z) = \|Z\|_*$ is restricted prox-regular, this is because $f_1$ is convex and each convex function is prox-regular [19], of course it is restricted prox-regular.

According to Theorem 1 in paper [22], our algorithm via ADMM will converge globally when the above five conditions of the objective function are satisfied. ■

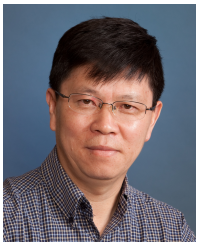
## REFERENCES


[1] Bianco A.M. and Yohai V.J. *Robust Estimation in the Logistic Regression Model*. Springer, 1996.
[2] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.
[3] JianFeng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2008.
[4] Emmanuel Candes, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58:1–37, 2009.
[5] J. B. Copas. Binary regression models for contaminated data. *Journal of the Royal Statistical Society*, 50(2):225–265, 1988.
[6] Jiashi Feng, Huan Xu, Shie Mannor, and Shuicheng Yan. Robust logistic regression and classification. In *Proceedings of NIPS*, pages 253–261, 2014.
[7] D. Freedman. *Statistical Models: Theory and Practice*. Cambridge University Press, 2009.
[8] Jonathan Gillard. An historical overview of linear regression with errors in both variables. *REVSTAT - Statistical Journal*, 8(1):57–80, 2010.
[9] Dong Huang, Ricardo Cabral, and De La Torre. Robust regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38:363–375, 2016.
[10] Hung Hung, ZhiYu Jou, and SuYun Huang. Robust mislabel logistic regression without modeling mislabel probabilities. *Biometrics.*, 74(1):145–154, 2018.
[11] Gbohounme I.L.T., Oscar Owino Ngesa, and Jude Eggoh. Self-selecting robust logistic regression model. *International Journal of Statistics and Probability*, 6(3):1–9, 2017.

[12] Kwangmoo Koh, Seung-Jean Kim, and Stephen P. Boyd. A method for large-scale $\ell_1$-regularized logistic regression. In *Proceedings of AAAI*, 2007.
[13] Sunin Lee, Honglak Lee, Pieter Abbeel, and Andrew Y Ng. efficient l1 regularized logistic regression. In *Proceedings of AAAI*, 2006.
[14] Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical report, 2009.
[15] Guangcan Liu and Ping Li. Low-rank matrix completion in the presence of high coherence. *IEEE Transactions on Signal Processing*, 64(21):5623–5633, 2016.
[16] Bogdan C. Matei and Peter Meer. Estimation of nonlinear errors-in-variables models for computer vision applications. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, 28(10):1537–1552, 2006.
[17] Tom M Mitchell. *Machine learning*. McGraw-Hill international editions - computer science series. McGraw-Hill Education, 1997.
[18] Heewon Park and Sadanori Konishi. Robust logistic regression modelling via the elastic net-type regularization and tuning parameter selection. *Journal of Statistical Computation and Simulation*, 86(7):1450–1461, 2016.
[19] R. A. Poliquin and R. T. Rockafellar. Prox-regular functions in variational analysis. *Transactions of the American Mathematical Society*, 348(5):1805–1838, 1996.
[20] Daryl Pregibon. Logistic regression diagnostics. *The Annals of Statistics*, 9(4):705–724, 1981.
[21] Soroosh Shafieezadeh-Abadeh, Peyman Mohajerin Esfahani, and Daniel Kuhn. Distributionally robust logistic regression. In *Proceedings of NIPS*, pages 1576–1584, 2015.
[22] Yu Wang, Wotao Yin, and Jinshan Zeng. Global convergence of admm in nonconvex nonsmooth optimization. *arXiv preprint arXiv:1511.06324*, 2015.
[23] John Wright, Arvind Ganesh, Shankar Rao, and Yi Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Proceedings of NIPS*, pages 2080–2088, 2009.
[24] Guibiao Xu, Bao Gang Hu, and Jose C. Principe. Robust bounded logistic regression in the class imbalance problem. In *Proceedings of IJCNN*, pages 1434–1441, 2016.
[25] Ming Yin, Junbin Gao, and Zhouchen Lin. Laplacian regularized low-rank representation and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):504–517, 2016.

**Ming Yin** (M'16-) received the Ph.D. degree in information and communication engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2006. He worked as a visiting scholar at the School of Computing and mathematics, Charles Sturt University, Bathurst, Australia, from Jan. 2012 to Dec. 2012. He is currently an Associate professor with the School of automation, Guangdong university of technology, Guangzhou, China. His research interests include computer vision, pattern recognition and machine learning. Ming Yin has served as the invited reviewer for IEEE TPAMI, IEEE TIP, IEEE TCYB, IEEE CVPR, AAAI, IEEE Access, IEEE TNNLS and Neurocomputing.

**Deyu Zeng** received the B.E. degree in Guangdong Polytechnic Normal University, Guangzhou, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Automation, Guangdong university of technology, Guangzhou, China.



**Junbin Gao** graduated from Huazhong University of Science and Technology (HUST), China in 1982 with BSc. degree in Computational Mathematics and obtained Ph.D. from Dalian University of Technology, China in 1991. He is a Professor of Big Data Analytics in the University of Sydney Business School at the University of Sydney and was a Professor in Computer Science in the School of Computing and Mathematics at Charles Sturt University, Australia. He was a senior lecturer, a lecturer in Computer Science from 2001 to 2005 at University of New England, Australia. From 1982 to 2001 he was an associate lecturer, lecturer, associate professor and professor in Department of Mathematics at HUST. His main research interests include machine learning, data analytics, Bayesian learning and inference, and image analysis.

**Zongze Wu** received his bachelor, master and doctor degree all in Xi'an Jiaotong University in 1999, 2002 and 2005, respectively. He worked in South China University of Technology from 2006 to 2016. He is currently a professor in School of automation, Guangdong University of Technology, China. He is the author of more than 50 papers, including 22 authorized patents. His research interests are concluded Control theory, Pattern recognition, IoT systems.

**Shengli Xie** (M'01-SM'02) received the M.S. degree in mathematics from Central China Normal University, Wuhan, China, in 1992, and the Ph.D. degree in automatic control from the South China University of Technology, Guangzhou, China, in 1997. Currently, he is the Director of both the Institute of Intelligent Information Processing (LI2P) and Guangdong Key Laboratory of IoT Information Technology, and also the professor of the School of Automation, Guangdong University of Technology, Guangzhou, China. He has authored or co-authored four monographs and more than 100 scientific papers published in journals and conference proceedings, and was granted more than 30 patents. He is a Senior member of the IEEE.