Robust Regression

Dong Huang, Ricardo Cabral and Fernando De la Torre, *Member, IEEE*

Abstract—Discriminative methods (*e.g.*, kernel regression, SVM) have been extensively used to solve problems such as object recognition, image alignment and pose estimation from images. These methods typically map image features ($\mathbf{X}$) to continuous (*e.g.*, pose) or discrete (*e.g.*, object category) values. A major drawback of existing discriminative methods is that samples are directly projected onto a subspace and hence fail to account for outliers common in realistic training sets due to occlusion, specular reflections or noise. It is important to notice that existing discriminative approaches assume the input variables $\mathbf{X}$ to be noise free. Thus, discriminative methods experience significant performance degradation when gross outliers are present.
Despite its obvious importance, the problem of robust discriminative learning has been relatively unexplored in computer vision. This paper develops the theory of Robust Regression (RR) and presents an effective convex approach that uses recent advances on rank minimization. The framework applies to a variety of problems in computer vision including robust linear discriminant analysis, regression with missing data, and multi-label classification. Several synthetic and real examples with applications to head pose estimation from images, image and video classification and facial attribute classification with missing data are used to illustrate the benefits of RR.

Index Terms—Robust methods, errors in variables, intra-sample outliers, missing data.

1 Introduction

Discriminative methods (*e.g.*, kernel regression, SVM) have been successfully applied to many computer vision problems. Unlike generative approaches, which produce a probability density over all variables, discriminative approaches directly attempt to compute the input to output mappings for classification or regression. Typically, discriminative models achieve better performance in classification tasks, especially when large amounts of training data are available. However, discriminative approaches often lack mathematically principled ways to incorporate priors. More importantly, existing discriminative models are not robust to errors in the data.

Linear and non-linear regression have been applied to solve a number of computer vision problems (*e.g.*, classification [1], pose estimation [2]). Although they are widely used, a major drawback of existing regression approaches is their lack of robustness to outliers and noise, which are common in realistic training sets due to occlusion, specular reflections or image noise. To better understand the lack of robustness, we consider the problem of learning a linear regressor from image features $\mathbf{X}$ to pose angles $\mathbf{Y}$ (see Fig. 1) by minimizing

$$\min_{\mathbf{T}} \|\mathbf{Y} - \mathbf{T}\mathbf{X}\|_F^2. \quad (1)$$

(see footnote[1] for an explanation of the notation used in this work). In the training stage, we learn the mapping $\mathbf{T}$, and in testing, we estimate the pose by projecting the features $\mathbf{x}_{te}$ of the test image, $\mathbf{T}\mathbf{x}_{te}$. Standard regression, Eq. (1), is optimal under the assumption that the error, $\mathbf{E} = \mathbf{Y} - \mathbf{T}\mathbf{X}$, is normally distributed. The Least Squares (LS) estimate is the most efficient unbiased estimate of $\mathbf{T}$ in the presence of Gaussian noise. This is the well-known Gauss-Markov theorem [3]. However, a small number of gross outliers can arbitrarily bias the estimate of the model's parameters ($\mathbf{T}$). It is important to note that in training and testing, $\mathbf{X}$ is assumed to be noise free. However, a single outlier in either training or testing can bias the projection because LS projects the data *directly* onto the subspace of $\mathbf{T}$. That is, the dot product of $\mathbf{x}_{te}$ with each row of $\mathbf{T}$ (*i.e.*, $\mathbf{T}\mathbf{x}_{te}$) can be largely biased by only a single outlier. For this reason, existing discriminative methods lack robustness to outliers.

The problem of robustness in regression has been studied thoroughly in statistics and recent decades have witnessed a fast-paced development of so-called robust methods (*e.g.*, [4], [5], [6]). For instance, M-estimators [4] assume the error has a heavy tail

• *Dong Huang, Ricardo Cabral and Fernando De La Torre are with the Robotics Institute, Carnegie Mellon University, PA, 15213, USA. E-mail: dghuang@andrew.cmu.edu, rscabral@cmu.edu and ftorre@cs.cmu.edu.*

1. Bold uppercase letters denote matrices ($\mathbf{D}$), bold lowercase letters denote column vectors (*e.g.*, $\mathbf{d}$). $\mathbf{d}_j$ represents the j^{th} column of the matrix $\mathbf{D}$. Non-bold letters represent scalar variables. $\|\mathbf{A}\|_F$ designates the Frobenius norm of matrix $\mathbf{A}$. $\|\mathbf{A}\|_*$ is the Nuclear norm (sum of singular values) of $\mathbf{A}$. The ℓ_0 norm of $\mathbf{A}$, $\|\mathbf{A}\|_0$, denotes the number of non-zero coefficients in $\mathbf{A}$. $\mathbf{I}_k \in \Re^{k\times k}$ denotes the identity matrix. $\mathbf{1}_n \in \Re^n$ is a vector of all ones. $\mathbf{0}_{k\times n} \in \Re^{k\times n}$ is a matrix of zeros. $\langle \mathbf{A}, \mathbf{B}\rangle$ denotes the inner product between two matrices $\mathbf{A}$ and $\mathbf{B}$. $\mathcal{S}_b(a) = \text{sgn}(a)\max(|a| - b, 0)$ denotes the shrinkage operator. $\mathcal{D}_\alpha(\mathbf{A})$ is the Singular Value Thresholding (SVT) operator, and the scalar α is a parameter of the SVT operator.

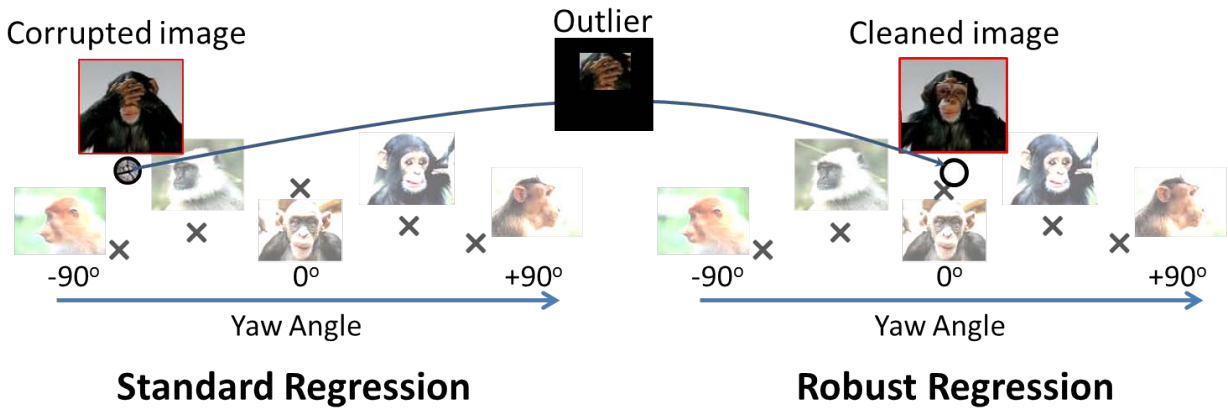


Fig. 1. Predicting the yaw angle of the monkey's head from image features. Note the image features (image pixels) contain outliers (hands of the monkey). (Left) Standard regression: projects a partially occluded frontal face image *directly* onto the head pose subspace and fails to estimate the correct yaw angle; (Right) Robust regression removes the intra-sample outlier and projects only the cleaned input image without biasing the yaw angle estimation.

and typically re-weight the whole sample inversely proportional to the error using different influence functions. That is, some robust approaches minimize a weighted regression $\sum_{i=1}^{n} w_i \|\mathbf{y}_i - \mathbf{T}\mathbf{x}_i\|_2^2$, where w_i weights the whole sample. Other robust approaches replace the sum (or the mean) by a more robust measure such as the median (*e.g.*, least median of squares) [7] or trimmed mean (*e.g.*, least trimmed square) [5]. However, all of the aforementioned traditional robust approaches for regression differ from the problem addressed in this paper in two ways: (1) these approaches do not model the error in $\mathbf{X}$ but in $\mathbf{Y} - \mathbf{T}\mathbf{X}$, (2) they mostly consider sample outliers (*i.e.*, the whole image is an outlier). This work proposes an intra-sample robust regression (RR) method that explicitly accounts for outliers in $\mathbf{X}$. Our work is related to errors in variables (EIV) models (*e.g.*, [8], [9], [10]). However, unlike existing EIV models, RR does not require a prior estimate of the noise and all parameters are automatically estimated.

In addition to reducing the influence of noise and outliers in regression, we extend RR to be able to deal with missing data in regression, wherein some elements of $\mathbf{X}$ are unknown. This is a common issue in computer vision applications since unknown elements typically correspond to unobserved local image features. Surprisingly, this problem has been relatively unexplored in the computer vision literature. We illustrate the power of RR in several computer vision tasks, including head pose estimation from images, facial attribute detection with missing data and robust LDA for multi-label image classification.

2 Related Work

Extensive literature exists on robust methods for regression. Huber [4] introduced M-estimation for regression, providing robustness to sample outliers. Rousseeuw and Leroy [5] proposed Least Trimmed Squares, which explicitly finds a data subset that minimizes the squared residual sum. Parallel to developments in the statistics community, the idea of subset selection has also flourished in many computer vision applications. Consensus approaches such as RANSAC [11] (and its Maximum Likelihood (ML) and M-estimator variants [12], [13]) randomly subsample input data to construct a tentative model. Model parameters are updated when a new configuration produces smaller inlier error than its predecessors. In spite of accurate parameter estimates, even in the presence of several outliers, these methods heavily rely on the assumption that model generation from a data subset is computationally inexpensive and inlier detection can be done adequately. Moreover, the aforementioned methods do not tackle *intra-sample* outliers, *i.e.*, partial sample corruptions.

To deal with noise in the variables, Error-In-Variable (EIV) approaches have been proposed (see [9] for an overview.) However, existing EIV approaches rely on strong parametric assumptions for the errors. For instance, orthogonal regression assumes that the variance of errors in the input and response variables are identical [14] or that their ratio is known [15]. Under these assumptions, orthogonal regression can minimize the Gaussian error orthogonal to the learned regression vectors. Grouping-based methods [16] assume that errors are respectively i.i.d. among the input and response variables so that one can split the data into groups and suppress the errors by computing either differences of the group sum, geometric means or instrument variables. Moment-based methods [17] learn the regression by estimating high-order statistics, *i.e.*, moments, from i.i.d. data. Likelihood-

based methods [10] learn a reliable regression when the input and response variables follow a joint, normal and identical distribution. Total Least Square (TLS) [9] and its nonlinear generalization [18] solve for additive/multiple terms that enforce the correlation between the input and response variables. TLS-based methods relax the assumptions in previous methods to allow correlated and non-identically distributed errors. Nevertheless, they still rely on parametric assumptions on the error. Unfortunately, in typical computer vision applications, errors caused by occlusion, shadow and edges seldom fit such distributions.

Although regression and classification are single-handedly modeled by our framework, several authors have addressed the issue of robust classification alone. The majority of these methods can be cast as robust extensions of Fisher/Linear Discriminant Analysis (FDA/LDA), where the empirical estimation of the class mean vectors and covariance matrices are replaced by their robust counterparts such as MVE estimators [19], MCD estimators [20] and S-estimators [21], [22]. In machine learning, several authors [23], [24] have proposed a worst-case FDA/LDA by minimizing the upper bound of the LDA cost function to increase the separation ability between classes under unbalanced sampling. As in previous work on robust regression, these methods are only robust to sample-outliers.

Our work is more related to recent work in computer vision. Fidler and Leonardis [25] incorporated robustness into LDA for intra-sample outliers. In the training stage, [25] computed PCA on the training data, replaced the minor PCA components by a robustly estimated basis, and then combined the two bases into a new one. Then, the data was projected onto the combined basis and LDA was computed. During testing, [25] first estimated the coefficients of test data on the recombined basis by sub-sampling the data elements using [26]. Finally, the class label of the test data was determined by applying learned LDA on the estimated coefficients. Although outliers outside of the PCA subspace can be suppressed, [25] does not address the problem of learning LDA with outliers in the PCA subspace of the training data. Zhu and Martinez [27] proposed learning an SVM with missing data that was robust to outliers. In [27], the possible values for missing elements are modeled by a Gaussian distribution such that for each class, the input data with all possible missing elements spans an affine subspace. The decision plane of the robust version of SVM jointly maximizes the between-class margin while minimizing the angle between the decision plane and the class-wise affine subspaces. However, [27] requires the location of the outliers to be known. In contrast to previous works, our RR enjoys several advantages: (1) it is a convex approach; (2) it does not impose assumptions, aside from sparsity, are imposed on the outliers, which makes our method general; (3) it automatically cleans the intra-sample outliers in the training data while learning a classifier.

Our work is inspired by existing work in robust PCA [28] and its recent advances due to rank minimization procedures [29], [30]. These methods model data as the sum of a low-rank clean data components with an arbitrary large and sparse outlier matrix. De La Torre and Black [28] increased PCA robustness by replacing the least-square metric with a robust function and re-weighted the influences of each component in each sample based on a given influence function (derivative of the robust function). Ke and Kanade [31] replaced the ℓ_2 norm with ℓ_1 to measure residuals between an input data matrix and its factorization and used an alternated linear programming to minimize it. [29], [30] separated a low-rank data matrix from an assumed sparse corruption despite its arbitrarily large magnitude and unknown pattern. A major advantage of this approach is the convex formulation. This approach has been extended to other problems such as background modeling and shadow removal [30], image tagging and segmentation [32], texture unwrapping [33] and segmentation [34]. These algorithms, however, were originally devised with tasks such as dimensionality reduction or matrix completion in mind, which are unsupervised in nature. In this paper, we will further extend the approach to detect intra-sample outliers in robust regression, and illustrate several applications in computer vision.

3 Robust Regression (RR)

This section describes the objective function for our proposed RR and its extension to robust LDA, as well as a detailed optimization algorithm for RR.

Let $\mathbf{X} \in \Re^{d_x \times n}$ be a matrix containing n d_x-dimensional samples possibly corrupted by outliers. Formally, $\mathbf{X} = \mathbf{D} + \mathbf{E}$, where $\mathbf{D} \in \Re^{d_x \times n}$ is a matrix containing the underlying noise-free component and $\mathbf{E} \in \Re^{d_x \times n}$ models outliers. In regression problems, one learns a mapping $\mathbf{T}$ from $\mathbf{X}$ to an output $\mathbf{Y} \in \Re^{d_y \times n}$. The outliers and the noise-free component $\mathbf{D}$ are unknown, so existing methods use $\mathbf{X}$ in the estimation of $\mathbf{T}$. In presence of outliers, this results in a biased estimation of $\mathbf{T}$. Our RR solves this problem by explicitly factorizing $\mathbf{X}$ into $\mathbf{D} + \mathbf{E}$ and only computing $\mathbf{T}$ using the cleaned data $\mathbf{D}$. RR solves the following optimization problem

$$\begin{aligned} \min_{\mathbf{T},\mathbf{D},\mathbf{E}} \quad & \frac{\eta}{2}\left\|\mathbf{W}(\mathbf{Y} - \mathbf{T}\hat{\mathbf{D}})\right\|_F^2 + rank(\mathbf{D}) + \lambda\|\mathbf{E}\|_0 \\ s.t. \quad & \mathbf{X} = \mathbf{D} + \mathbf{E}, \quad \hat{\mathbf{D}} = [\mathbf{D}; \mathbf{1}^T], \end{aligned} \tag{2}$$

where $\mathbf{W} \in \Re^{d_y \times d_y}$ is a diagonal matrix that weights the output dimensions, $\mathbf{T} \in \Re^{(d_x+1) \times d_y}$ is the regression matrix (the extra dimension is for the regression bias term). η and λ are scalars that weight the first and third term in Eq. (2) respectively. RR explicitly avoids projecting the outlier matrix $\mathbf{E}$ to the output

Algorithm 1 ALM algorithm for solving RR Eq. (3)

Require: $\mathbf{X}$, $\mathbf{Y}$, parameters η (a positive scalar weights term $\left\|\mathbf{W}(\mathbf{Y}-\mathbf{T}\hat{\mathbf{D}})\right\|_F^2$), λ (a positive scalar weights term $\|\mathbf{E}\|_1$), ρ (a positive scalar for updating the Lagrange coefficients), γ (a positive scalar for regularizing the solution to $\mathbf{T}$).

Initialization:$\mathbf{D}^{(0)}=\mathbf{X}$, $\hat{\mathbf{D}}^{(0)}=[\mathbf{D}^{(0)};\mathbf{1}^T]$, $\mathbf{E}^{(0)}=\mathbf{X}-\mathbf{D}^{(0)}$, $\mathbf{T}^{(0)}=(\hat{\mathbf{D}}^{(0)}(\hat{\mathbf{D}}^{(0)})^{\mathbf{T}}+\gamma\mathbf{I}_{d_x+1})^{-1}\mathbf{Y}(\hat{\mathbf{D}}^{(0)})^T$;
Lagrange Multiplier Initialization: $\Gamma_1^{(0)}=\frac{\mathbf{X}}{\|\mathbf{X}\|_2}$,$\Gamma_2^{(0)}=\frac{\mathbf{D}^{(0)}}{\|\mathbf{D}^{(0)}\|_2}$,$\mu_1^{(0)}=\frac{dn}{4}\|\mathbf{X}\|_1$, $\mu_2^{(0)}=\frac{dn}{4}\|\mathbf{D}^{(0)}\|_1$.
while $\frac{\|\mathbf{X}-\mathbf{D}^{(k)}-\mathbf{E}^{(k)}\|_F}{\|\mathbf{X}\|_F}>10^{-8}$ and $\frac{\|\hat{\mathbf{D}}^{(k)}-[\mathbf{D}^{(k)};\mathbf{1}^T]\|_F}{\|\hat{\mathbf{D}}^{(k)}\|_F}>10^{-8}$ **do**
Assuming $\mathbf{W}=diag\{w_{ii}\}$, update $\mathbf{T}^{(k+1)}=[\mathbf{t}_1,\mathbf{t}_2,\cdots,\mathbf{t}_c]$, where $\mathbf{t}_i=w_{ii}^2(w_{ii}^2\hat{\mathbf{D}}^{(k+1)}(\hat{\mathbf{D}}^{(k+1)})^T+\gamma\mathbf{I}_d)^{-1}\mathbf{y}_i(\hat{\mathbf{D}}^{(k)})^T$, and γ regularizes the scale of $\mathbf{t}_i$.
Update $\hat{\mathbf{D}}^{(k+1)}=\left[\eta(\mathbf{T}^{(k)})^T\mathbf{W}^T\mathbf{W}\mathbf{T}^{(k)}+\mu_2^{(k)}\mathbf{I}_d\right]^{-1}\left[\eta(\mathbf{T}^{(k)})^T\mathbf{W}^T\mathbf{Y}-\Gamma_2^{(k)}+\mu_2^{(k)}[\mathbf{D}^{(k)};\mathbf{1}^T]\right]$;
Update $\mathbf{D}^{(k+1)}=\mathcal{D}_{1/\beta}(\mathbf{Z}^{(k+1)})$, where $\mathbf{Z}^{(k+1)}=\frac{1}{\beta}\left(\Gamma_1^{(k)}+\mu_1^{(k)}\left(\mathbf{X}-\mathbf{E}^{(k)}\right)+\left[\Gamma_2^{(k)}+\mu_2^{(k)}\hat{\mathbf{D}}^{(k)}\right]_{(1:d_x,\cdot)}\right)$, and $\beta=\mu_1^{(k)}+\mu_2^{(k)}$;
Update $\mathbf{E}^{(k+1)}=\mathcal{S}_{\lambda/\mu_1^{(k)}}\left(\mathbf{X}-\mathbf{D}^{(k)}+\Gamma_1^{(k)}/\mu_1^{(k)}\right)$;
Update $\Gamma_1^{(k+1)}=\Gamma_1^{(k)}+\mu_1^{(k+1)}(\mathbf{X}-\mathbf{D}^{(k+1)}-\mathbf{E}^{(k+1)})$, $\Gamma_2^{(k+1)}=\Gamma_2^{(k)}+\mu_2^{(k+1)}(\hat{\mathbf{D}}^{(k+1)}-[\mathbf{D}^{(k+1)};\mathbf{1}^T])$, $\mu_1^{(k+1)}=\rho\mu_1^{(k)}$, $\mu_2^{(k+1)}=\rho\mu_2^{(k)}$;
end while
Ensure: $\mathbf{T}$, $\mathbf{D}$, $\mathbf{E}$

space by learning the regression $\mathbf{T}$ only from the augmented noise-free data $\hat{\mathbf{D}}=[\mathbf{D};\mathbf{1}^T]\in\Re^{(d_x+1)\times n}$. Note that there are infinite possible decompositions of $\mathbf{X}$ into $\mathbf{D}$ and $\mathbf{E}$. RR thus adds the second and third terms in Eq. (2) to constrain the possible solutions. The second term constrains $\mathbf{D}$ to lie in a low-dimensional subspace, which is a good prior for visual data. The third term encourages $\mathbf{E}$ to be sparse.

It is important to note that RR is different from cleaning the data using RPCA and then computing LS-regression on the clean data, because RR *cleans* the input data $\mathbf{X}=\mathbf{D}+\mathbf{E}$ in a supervised manner. That is, the data $\mathbf{D}$ will preserve the subspace of $\mathbf{X}$ that is maximally correlated with $\mathbf{Y}$. For this reason, the outlier component $\mathbf{E}$ computed by RR is able to correct outliers both inside and outside the subspace spanned by $\mathbf{D}$ (see the experiment in section 5.1.1).

The original form of RR, Eq. (2), is cumbersome to solve because the rank and cardinality operators are discontinuous and non-convex. Following recent advances on rank minimization [30], these operators are respectively relaxed to their convex surrogates: the nuclear norm and the ℓ_1-norm. Using this relaxation Eq. (2) is rewritten as

$$\begin{aligned}\min_{\mathbf{T},\mathbf{D},\mathbf{E}}\quad & \frac{\eta}{2}\left\|\mathbf{W}(\mathbf{Y}-\mathbf{T}\hat{\mathbf{D}})\right\|_F^2+\|\mathbf{D}\|_*+\lambda\|\mathbf{E}\|_1\\ s.t.\quad & \mathbf{X}=\mathbf{D}+\mathbf{E},\hat{\mathbf{D}}=[\mathbf{D};\mathbf{1}^T].\end{aligned}\tag{3}$$

This problem can be efficiently optimized using an Augmented Lagrange Muliplier (ALM) technique, wherein Eq. (3) is rewritten as

$$\begin{aligned}\min_{\mathbf{T},\mathbf{D},\hat{\mathbf{D}},\mathbf{E}}\quad & \frac{\eta}{2}\left\|\mathbf{W}(\mathbf{Y}-\mathbf{T}\hat{\mathbf{D}})\right\|_F^2+\|\mathbf{D}\|_*+\lambda\|\mathbf{E}\|_1\\ & +\langle\Gamma_1,\mathbf{X}-\mathbf{D}-\mathbf{E}\rangle+\frac{\mu_1}{2}\|\mathbf{X}-\mathbf{D}-\mathbf{E}\|_F^2\\ & +\langle\Gamma_2,\hat{\mathbf{D}}-[\mathbf{D};\mathbf{1}^T]\rangle+\frac{\mu_2}{2}\|\hat{\mathbf{D}}-[\mathbf{D};\mathbf{1}^T]\|_F^2,\end{aligned}\tag{4}$$

where $\Gamma_1\in\Re^{d_x\times n}$ and $\Gamma_2\in\Re^{(d_x+1)\times n}$ are Lagrange multiplier matrices, and μ_1 and μ_2 are the penalty parameters. For each of the four matrices $\{\mathbf{T},\mathbf{D},\hat{\mathbf{D}},\mathbf{E}\}$ to be solved in Eq. (4), the cost function is convex if the remainder three matrices are kept fixed. Details of the ALM method to minimize Eq. (4) are given in Alg. 1.

3.1 Robust LDA: Extending RR for classification

Classification problems can be cast as a particular case of binary regression, where each sample in $\mathbf{X}$ belongs to one of c classes. The goal is then to learn a mapping from $\mathbf{X}$ to labels indicating the class membership of the data points. LDA learns a linear transformation that maximizes inter-class separation while minimizing intra-class variance, and typical solutions are based on solving a generalized eigenvalue problem. However, when learning from high-dimensional data such as images ($n<d_x$), LDA typically suffers from the small sample size problem. While there are several approaches to solve the small sample size problem (*e.g.*, regularization), a more fundamental solution is to relate the LDA problem to a reduced-rank LS problem [35]. LS-LDA [35] directly maps $\mathbf{X}$ to the

class labels by minimizing

$$\min_{\mathbf{T}} \left\| (\mathbf{Y}\mathbf{Y}^T)^{-1/2}(\mathbf{Y} - \mathbf{T}\mathbf{X}) \right\|_F^2, \quad (5)$$

where $\mathbf{Y} \in \Re^{c \times n}$ is a binary indicator matrix, such that $y_{ij} = 1$ if $\mathbf{x}_i$ belongs to class j, otherwise $y_{ij} = 0$. The normalization factor $\mathbf{W} = (\mathbf{Y}\mathbf{Y}^T)^{-1/2}$ compensates for having a different number of samples per class. $\mathbf{T} \in \Re^{c \times d_x}$ is a reduced rank regression matrix, which typically has rank $c - 1$ (if the data are centered). After $\mathbf{T}$ is learned, a test data sample $\mathbf{x}_{te} \in \Re^{d_x \times 1}$ is projected by $\mathbf{T}$ onto the c dimensional output space spanned by $\mathbf{T}$, then the class label of the test data $\mathbf{x}_{te}$ is assigned using k-NN.

When $\mathbf{X}$ is corrupted by outliers, Eq. (5) suffers from the same bias problem as standard regression. RR, Eq. (3), can be directly applied to Eq. (5), yielding

$$\begin{aligned} \min_{\mathbf{T},\mathbf{D},\mathbf{E}} \quad & \frac{\eta}{2} \left\| (\mathbf{Y}\mathbf{Y}^T)^{-1/2}(\mathbf{Y} - \mathbf{T}\hat{\mathbf{D}}) \right\|_F^2 + \|\mathbf{D}\|_* + \lambda \|\mathbf{E}\|_1 \\ s.t. \quad & \mathbf{X} = \mathbf{D} + \mathbf{E}, \quad \hat{\mathbf{D}} = [\mathbf{D}; \mathbf{1}^T]. \end{aligned} \quad (6)$$

This is a Robust LDA formulation, which can be easily solved as a special case of RR (Alg. 1).

3.2 Robustness in testing

In the previous sections, we have assumed that the training set was corrupted by outliers and noise. Similarly, the test data might contain outliers and, as in the case of training, RR removes outliers before projection. Let us refer to $\mathbf{X}_{te} \in \Re^{d_x \times n_{te}}$ as a set of test samples (n_{te} samples), $\mathbf{Y}_{te} \in \Re^{d_y \times n_{te}}$ as the estimated label, and let the subscript te denote the test data. Note that this is a non-trivial problem because the test label matrix $\mathbf{Y}_{te}$ is not available to provide the supervised information.

Consider Eq. (3) without the first supervised term,

$$\begin{aligned} \min_{\mathbf{D}_{te},\mathbf{E}_{te}} \quad & \|\mathbf{D}_{te}\|_* + \lambda \|\mathbf{E}_{te}\|_1 \\ s.t. \quad & \mathbf{X}_{te} = \mathbf{D}_{te} + \mathbf{E}_{te}, \end{aligned} \quad (7)$$

where $\mathbf{D}_{te} \in \Re^{d_x \times n_{te}}$ is the cleaned test data, $\mathbf{E}_{te} \in \Re^{d_x \times n_{te}}$ is the noise/outlier matrix, and λ is the positive scalar determined in training (see Eq. (3)).

Eq. (7) is equivalent to RPCA [29]. However, RPCA is an unsupervised technique and can only clean outliers/noise that are orthogonal to $\mathbf{X}_{te}$. We will refer to this noise as out-of-subspace noise. If we are interested in removing the error within the subspace of $\mathbf{X}_{te}$, this can be done by using the cleaned training data $\mathbf{D}$. In the training stage, $\mathbf{D}$ is optimized to have maximum correlation with the output labels $\mathbf{Y}$. Our assumption is that the clean test data can be reconstructed as local combinations of the training data. That is, $\mathbf{D}_{te} = \mathbf{D}\mathbf{Z}_{te}$, where $\mathbf{Z}_{te} \in \Re^{n \times n_{te}}$. In order to make the combination locally compact, we regularize the combination coefficient $\mathbf{Z}_{te}$ by minimizing its nuclear norm [36]. The resulting objective function becomes

$$\begin{aligned} \min_{\mathbf{Z}_{te},\mathbf{E}_{te}} \quad & \|\mathbf{Z}_{te}\|_* + \frac{\lambda}{\|\mathbf{D}\|_*} \|\mathbf{E}_{te}\|_1 \\ s.t. \quad & \mathbf{X}_{te} = \mathbf{D}\mathbf{Z}_{te} + \mathbf{E}_{te}, \end{aligned} \quad (8)$$

where the weight $\frac{\lambda}{\|\mathbf{D}\|_*}$ in front of $\|\mathbf{E}_{te}\|$ is used to keep the original balance between $\|\mathbf{E}_{te}\|$ and $\|\mathbf{D}_{te}\| = \|\mathbf{D}\mathbf{Z}_{te}\|$ in Eq. (7). Directly applying the ALM to solve Eq. (8) is a challenging task because we cannot apply the standard Singular Value Thresholding (SVT) operators on $\mathbf{Z}_{te}$. Note that the term $\mathbf{D}\mathbf{Z}_{te}$ is not the standard formulation to be solved with SVT. We followed the idea of [37] and linearized the term $\mathbf{D}\mathbf{Z}_{te}$ before the standard SVT operation. Alg. 2 describes the optimization strategy. After solving (8), the regression or classification output for $\mathbf{X}_{te}$ is computed as $\mathbf{Y}_{te} = \mathbf{T}[\mathbf{D}\mathbf{Z}_{te}; \mathbf{1}^T]$. In the case of classification, $\mathbf{Y}_{te}$ contains the decision values to compute AUROC or to produce binary class labels using the k-nearest-neighbor method.

4 RR WITH MISSING DATA

Robust regression Eq. (3) can easily be extended to handle missing elements in the input data matrix $\mathbf{X}$. From now on, we will refer to this problem as "RR-Missing".

Let Ω be the index set of observed elements in $\mathbf{X}$, and $\mathcal{P}_\Omega$ be the projection operator from the matrix space to the support of observed elements. RR-Missing solves the following problem

$$\begin{aligned} \min_{\mathbf{T},\mathbf{D},\mathbf{E}} \quad & \frac{\eta}{2} \left\| \mathbf{W}(\mathbf{Y} - \mathbf{T}\hat{\mathbf{D}}) \right\|_F^2 + \|\mathbf{D}\|_* + \lambda \|\mathbf{E}\|_1 \\ s.t. \quad & \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{D} + \mathbf{E}), \;\; \hat{\mathbf{D}} = [\mathbf{D}; \mathbf{1}^T], \end{aligned} \quad (9)$$

The algorithm for solving Eq. (9) is similar to Eq. (3). After solving Eq. (9), the missing elements in $\mathbf{X}$ are filled by the values in $\mathbf{D}$.

As in the case of RR, the test data with missing elements can be cleaned similarly to section 3.2 by solving

$$\begin{aligned} \min_{\mathbf{Z}_{te},\mathbf{E}_{te}} \quad & \|\mathbf{Z}_{te}\|_* + \frac{\lambda}{\|\mathbf{D}\|_*} \|\mathbf{E}_{te}\|_1 \\ s.t. \quad & \mathcal{P}_\Omega(\mathbf{X}_{te}) = \mathcal{P}_\Omega(\mathbf{D}\mathbf{Z}_{te} + \mathbf{E}_{te}). \end{aligned} \quad (10)$$

After solving Eq. (10), the regression/classification output for $\mathbf{X}_{te}$ is computed as $\mathbf{Y}_{te} = \mathbf{T}[\mathbf{D}\mathbf{Z}_{te}; \mathbf{1}^T]$. The extension of RR-Missing to RLDA-Missing is straightforward.

5 EXPERIMENTAL RESULTS

This section compares our RR methods against state-of-the-art approaches on four experiments for regression and classification.

Algorithm 2 ALM algorithm for cleaning the test data Eq. (8)

Require: $\mathbf{X}_{te} \in \Re^{d_x \times n_{te}}$, $\mathbf{D} \in \Re^{d_x \times n}$, parameters λ (a positive scalar weights term $\|\mathbf{E}\|_1$, which is determined in training) and ρ_t (a positive scalar for updating the Lagrange coefficients).

Initialization:$\mathbf{Z}_{te}^{(0)} = \mathbf{0}_{n \times n_{te}}$, where its element $\mathbf{z}_{te}^{(0)}(i,j) = 1$ if $i = arg\min_i\{dist(\mathbf{x}_{te}(j), \mathbf{d}_i)\}_{i=1,\cdots,n} j = 1, \cdots, n_{te}$; $\mathbf{E}_{te}^{(0)} = \mathbf{X}_{te} - \mathbf{D}\mathbf{Z}_{te}^{(0)}$;
Lagrange Multiplier Initialization: $\Gamma_{te}^{(0)} = \frac{\mathbf{X}_{te}}{\|\mathbf{X}_{te}\|_F}$,$\mu_{te}^{(0)} = \frac{dn}{4}\|\mathbf{X}_{te}\|_1$.
while $\frac{\left\|\mathbf{X}_{te} - \mathbf{D}\mathbf{Z}_{te}^{(k)} - \mathbf{E}_{te}^{(k)}\right\|_F}{\|\mathbf{X}_{te}\|_F} > 10^{-8}$ **do**
  Update $\mathbf{S}^{(k+1)} = \mathbf{Z}_{te}^{(k)} - \frac{1}{\beta_{te}}\left(-\mathbf{D}^T\Gamma^{(k)} + \mu_{te}^{(k)}\mathbf{D}^T\left[\mathbf{D}\mathbf{Z}_{te}^{(k)} - (\mathbf{X} - \mathbf{E}_{te}^{(k)})\right]\right)$, where $\beta_{te} = \mu_{te}^{(k)}\|\mathbf{D}^T\mathbf{D}\|_F^2$;
  Update $\mathbf{Z}_{te}^{(k+1)} = \mathcal{D}_{1/\beta}(\mathbf{S}^{(k+1)})$;
  Update $\mathbf{E}_{te}^{(k+1)} = \mathcal{S}_{\lambda/\mu_{te}^{(k)}}\left(\mathbf{X}_{te} - \mathbf{D}\mathbf{Z}_{te}^{(k)} + \Gamma_{te}^{(k)}/\mu_{te}^{(k)}\right)$;
  Update $\Gamma_{te}^{(k+1)} = \Gamma_{te}^{(k)} + \mu_{te}^{(k)}(\mathbf{X} - \mathbf{D}\mathbf{Z}_{te}^{(k+1)} - \mathbf{E}_{te}^{(k+1)})$, $\mu_{te}^{(k+1)} = \rho_t\mu_{te}^{(k)}$;
end while
Ensure: $\mathbf{Z}_{te}$, $\mathbf{E}_{te}$

The first experiment uses synthetic data to compare with existing approaches and illustrate how existing robust regression methods cannot remove outliers that lie in the subspace of the data. The second experiment applies RR to the problem of head pose estimation from partially corrupted images. The third experiment reports comparisons of RR against state-of-the-art multi-label classification algorithms on the MSRC, Mediamill and TRECVID2011 databases. The fourth experiment illustrates the application of RR-Missing to predict facial attributes.

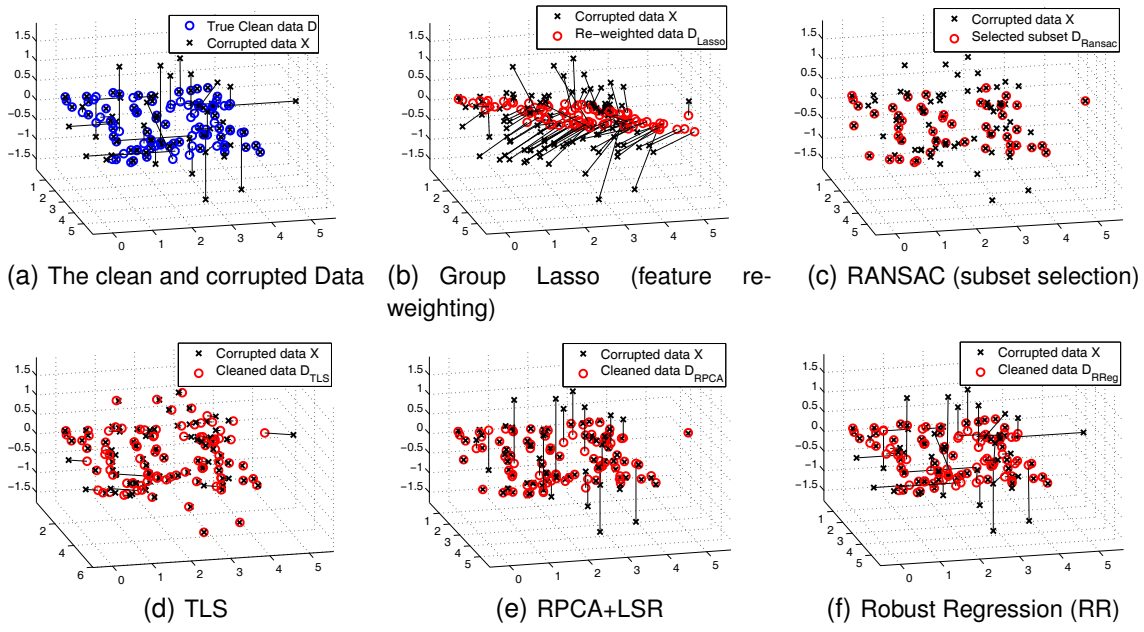
5.1 Robust Regression (RR)

5.1.1 RR on Synthetic Data

This section illustrates the benefits of RR in a synthetic example. We generated 200 three-dimensional samples where the first two components were generated from a uniform distribution between $[0, 6]$, and the third dimension is 0. In Matlab notation $\mathbf{D} = [6 * rand(2, 200); \mathbf{0}^T]$, , $\mathbf{X} = \mathbf{D} + \mathbf{E}$, , and $\mathbf{Y} = \mathbf{T}_*[\mathbf{D}; \mathbf{1}^T]$, where $\mathbf{D} \in \Re^{3\times 200}$ is the clean data. $\mathbf{T}_* \in \Re^{3\times 4}$ was randomly generated and used as the true regression matrix. The error term, $\mathbf{E} \in \Re^{3\times 200}$, was generated as follows: for 20 random samples, we added random Gaussian noise ($\sim \mathcal{N}(0, 1)$) in the second dimension, this simulates in-subspace noise. Similarly, for another 20 random samples, we added random Gaussian noise ($\sim \mathcal{N}(0, 1)$) to the third dimension. This simulates noise outside the subspace. The output data matrix was generated as $\mathbf{Y} = \mathbf{T}_*[\mathbf{D}; \mathbf{1}^T] \in \Re^{3\times 200}$. Fig. 2 (a) shows the clean data $\mathbf{D}$ with blue "o's", and the corrupted data $\mathbf{X}$ with black "×'s". For better visualization, we only showed 100 randomly selected samples. The black line segments connect the same samples before ($\mathbf{D}$) and after corruption ($\mathbf{X}$). The line segments along the vertical direction are the out-of-subspace component of $\mathbf{E} = \mathbf{X} - \mathbf{D}$, and the horizontal line segments represent the in-subspace component of $\mathbf{E}$.

We compared our RR with five state-of-the-art methods: (1) Standard least-squares regression (LSR), (2) GroupLasso (GLasso) [38], (3) RANSAC [11], (4) Total Least Square (TLS) [39], which assumes the error in the data is additive and follows a Gaussian distribution, and (5) RPCA+LSR, which consists of first performing RPCA [29] on the input data and then learning the regression on the cleaned data using standard LSR. The LSR directly learns the regression matrix $\mathbf{T}$ using the data $\mathbf{X}$. The other methods (2)-(5) re-weight the data or select a subset of the samples input data $\mathbf{X}$ before learning the regression. We randomly selected 100 samples for training and used the remaining 100 data points for testing. Both the training and testing sets contain half of the corrupted samples.

Fig. 2(b-f) visualizes the results of the regression for the different methods. Fig. 2(b) shows the results of $\mathbf{TX}$, once $\mathbf{T}$ is learned with GLasso. GLasso learns a sparse regression matrix that re-weights the input data along dimensions, but it is unable to handle intra-sample outliers. Note how the samples are far away from the original clean samples. Fig. 2(c) shows the subset of $\mathbf{X}$ selected by RANSAC. Although we selected RANSAC parameters to obtain the best testing error, many of the corrupted data points are still identified as inliers. Fig. 2 (d) shows results obtained by TLS, where TLS only partially cleaned the corrupted data because the synthesized error cannot be modeled by an isotropic Gaussian distribution. Fig. 2 (e) shows results obtained by the method RPCA+LSR, which first computes RPCA to clean the data and then applies LSR. The data cleaned by RPCA [29], $\mathbf{D}_{RPCA}$, is displayed with red "o's". Because $\mathbf{D}_{RPCA}$ is computed in an unsupervised manner, only the out-of-subspace error (the vertical lines) can be discarded, while the in-subspace outliers can not be corrected.



(a) The clean and corrupted Data (b) Group Lasso (feature re-weighting) (c) RANSAC (subset selection)

(d) TLS (e) RPCA+LSR (f) Robust Regression (RR)

Fig. 2. (a) Original and corrupted 3D synthetic dataset. Black lines connect data points before ($\mathbf{D}$) and after corruption ($\mathbf{X}$). (b)-(e) show the input data processed by several baselines, and (f) shows that RR removes the in-subspace outliers.

TABLE 1
Relative Absolute Error (RAE) and its standard deviation for output $\mathbf{Y}_{te}$ and regression matrix $\mathbf{T}$ on synthetic data (10 repetitions).

	LSR	GLasso	RANSAC	TLS	RPCA+LSR	**RR**
$RAE_{\mathbf{T}}$	0.269 ± 0.121	0.269 ± 0.121	0.256 ± 0.133	0.269 ± 0.121	0.464 ± 0.030	$\mathbf{0.035 \pm 0.015}$
$RAE_{\mathbf{Y}}$	0.035 ± 0.012	0.035 ± 0.012	0.036 ± 0.013	0.925 ± 0.136	0.051 ± 0.006	$\mathbf{0.015 \pm 0.006}$

Finally, Fig. 2 (f) shows the result of RR. The clean data $\mathbf{D}_{RR}$ is denoted by red "o's". Note that our approach is able to clean both the in-subspace (the horizontal lines) and out-of-subspace (the vertical lines) outliers. This is because our method jointly computes the regression and the subspace estimation.

We also computed the error for the regression matrix $\mathbf{T}_*$ (the first two columns) and the testing error for $\mathbf{Y}_{te}$ on the 100 test samples. Table 1 compares the mean regression error measured by the Relative Absolute Error (RAE) between the true labels $\mathbf{Y}_{te} \in \Re^{3\times 100}$ and the estimated labels $\widetilde{\mathbf{Y}_{te}}$. $RAE_{\mathbf{T}} = \frac{\|\mathbf{T}(:,1:2)-\mathbf{T}_*(:,1:2)\|_F}{\|\mathbf{T}_*(:,1:2)\|_F}$ and $RAE_{\mathbf{Y}} = \frac{\|\widetilde{\mathbf{Y}_{te}}-\mathbf{Y}_{te}\|_F}{\|\mathbf{Y}_{te}\|_F}$. The information in the third column of $\mathbf{T}_*$ is excluded in generating $\mathbf{Y} = \mathbf{T}[\mathbf{D}; \mathbf{1}^T]$. Therefore, we dismiss this column when evaluating $RAE_{\mathbf{T}}$. As shown in Table 1, RR produces the smallest estimation error for both $\mathbf{T}_*$ and $\mathbf{Y}_{te}$ among the five compared methods, while GroupLasso, RANSAC and RPCA+LSR produce small improvements over standard LSR due to their limitability to deal with both the in-subspace and out-of-subspace corruptions.

5.1.2 RR for pose estimation

This section illustrates the benefit of RR in the problem of head pose estimation. We used a subset of the CMU Multi-PIE database [40], which contains 1721 face images from 249 subjects in Session 1. The face regions are detected automatically using the OpenCV [2] face detector. The detected faces cover 11 head poses θ =[$-90°$, $-75°$, $-60°$, $-45°$, $-15°$, $0°$, $15°$, $45°$, $60°$, $75°$, $90°$] each with a random lighting direction. Each image is cropped around the face region and resized to 51×61. We vectorized the images into a vector of $51 \times 61 = 3111$ dimensions in the matrix $\mathbf{X} \in \Re^{3111\times 1721}$ and the yaw angles of the images are used as the output data $\mathbf{Y} = [\cos(\theta), \sin(\theta)] \in \Re^{2\times 1721}$. See Fig. 4 for examples of cropped images.

Similar to the previous section, we have compared RR with five methods to learn a regression from the image $\mathbf{X}$ to the yaw angle $\mathbf{Y}$: (1) LSR, (2) GLasso [38], (3) RANSAC [11], (4) TLS and (5) RPCA+LSR. For a fair comparison, we randomly divided the 249 subjects into 5 folds and performed 5-fold cross-validation. For each trial of cross-validation, we used one fold for training and the remaining four folds for testing. Parameters of interest in methods (2)-(4) were selected by performing grid search over the 5-fold cross-validation. The performance of the compared methods is measured with the averaged angle error.

2. http://opencv.willowgarage.com/wiki/

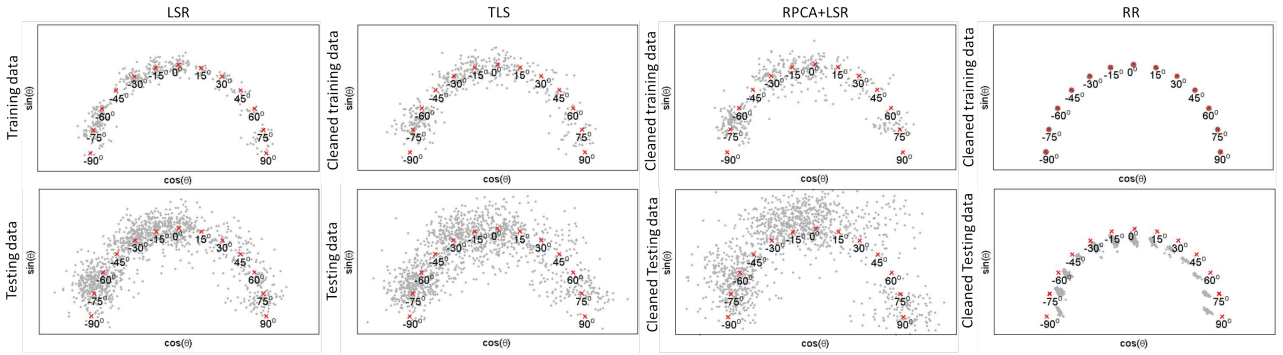
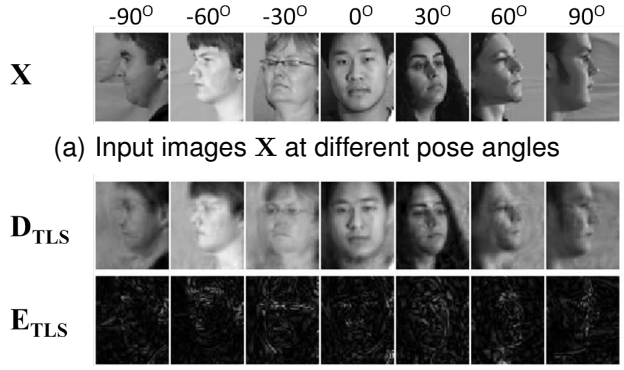


Fig. 3. Projection of face images (the gray "·'s")in the output space $\mathbf{Y} = [\cos(\theta), \sin(\theta)]$ by LSR, TLS, RPCA+LSR and Robust Regression(RR). The red "×'s" denote the ground true location for pose angles θ =[$-90°$, $-75°$, $-60°$, $-45°$, $-15°$, $0°$, $15°$, $45°$, $60°$, $75°$, $90°$] in the output space.



(a) Input images $\mathbf{X}$ at different pose angles

(b) Decomposition of images in (a) as $\mathbf{X} = \mathbf{D}_{TLS} + \mathbf{E}_{TLS}$ by TLS

(c) Decomposition of images in (a) as $\mathbf{X} = \mathbf{D}_{RPCA} + \mathbf{E}_{RPCA}$ by RPCA

(d) Decomposition of images in (a) as $\mathbf{X} = \mathbf{D}_{RR} + \mathbf{E}_{RR}$ by RR.

Fig. 4. Decomposition of input images in (a) by (b) TLS, (c) RPCA and (d) RR. Robust regression (RR) cleans most facial details and only preserves the correlated with pose angles.

Table 2 summarizes the results of methods (1)-(4) and RR. The LSR method produced the largest angle error. RANSAC produced comparable error to standard LSR, indicating that RANSAC is unable to select a subset of "inliers" to robustly estimate the regression matrix. RPCA+LSR produced relatively larger yaw angle error. This is because RPCA is unsupervised and lacks the ability to preserve the discriminative information in $\mathbf{X}$ that correlates with the angles $\mathbf{Y}$. RR got the smallest error among all the compared methods.

To further illustrate how RR differs from TLS and RPCA+LSR, Fig. 4 visualizes the decomposition of training images by RR (*i.e.*, $\mathbf{X} = \mathbf{D}_{RR} + \mathbf{E}_{RR}$), by TLS (*i.e.*, $\mathbf{X} = \mathbf{D}_{TLS} + \mathbf{E}_{TLS}$) and by RPCA (*i.e.*, $\mathbf{X} = \mathbf{D}_{RPCA} + \mathbf{E}_{RPCA}$), for the same input images. All images contain person-specific features, for instance glasses at $-30°$ and long dark hair at $+30°$ (see Fig. 4(a)). Fig. 4(b)-(c) show that both TLS and RPCA are able to remove some of the edges. While RR (Fig. 4(d)) preserves much fewer personal facial details in $\mathbf{D}_{RR}$ than TLS ($\mathbf{D}_{TLS}$) and RPCA ($\mathbf{D}_{RPCA}$) (especially for those images under the pose $-30°$ and $+30°$). With fewer facial details and more dominant profiles, the regression trained on $\mathbf{D}_{RR}$ (as in RR) is able to model higher correlation with the pose angles than using $\mathbf{D}_{RPCA}$.

Fig. 3 visualizes the differences among LSR, TLS, RPCA+LDA and RR on both training (the 1^{st} row) and testing images (the 2^{nd} row). We projected the face images (the gray "·'s") into the output space $\mathbf{Y} = [\cos(\theta), \sin(\theta)]$ using the discussed four methods (one column each). The red "×'s" denote the ground true location for pose angles. The projections (the gray "·'s") produced by LSR, TLS and RPCA+LDA are far from the ideal outputs (the red "×'s"). RR (the 4^{th} column) is the method that improves the correlation between inputs (the gray "·'s") and the outputs (the red "×'s"). It is therefore more robust than LSR, TLS and RPCA+LSR in estimating the pose angles.

5.2 Robust LDA for classification

5.2.1 *RLDA for face recognition*

This section evaluates our Robust LDA (RLDA) method for face recognition with synthetically cor-

TABLE 2
Comparison of yaw angle error and standard deviation for six methods on a subset of CMU Multi-PIE database [40].

LSR	GLasso	RANSAC	TLS	RPCA+LSR	**RR**
$7.3^o \pm 6.1^o$	$7.1^o \pm 5.9^o$	$7.3^o \pm 6.2^o$	$11.7^o \pm 10.1^o$	$10.8^o \pm 9.7^o$	$\mathbf{5.1}^o \pm \mathbf{4.6}^o$

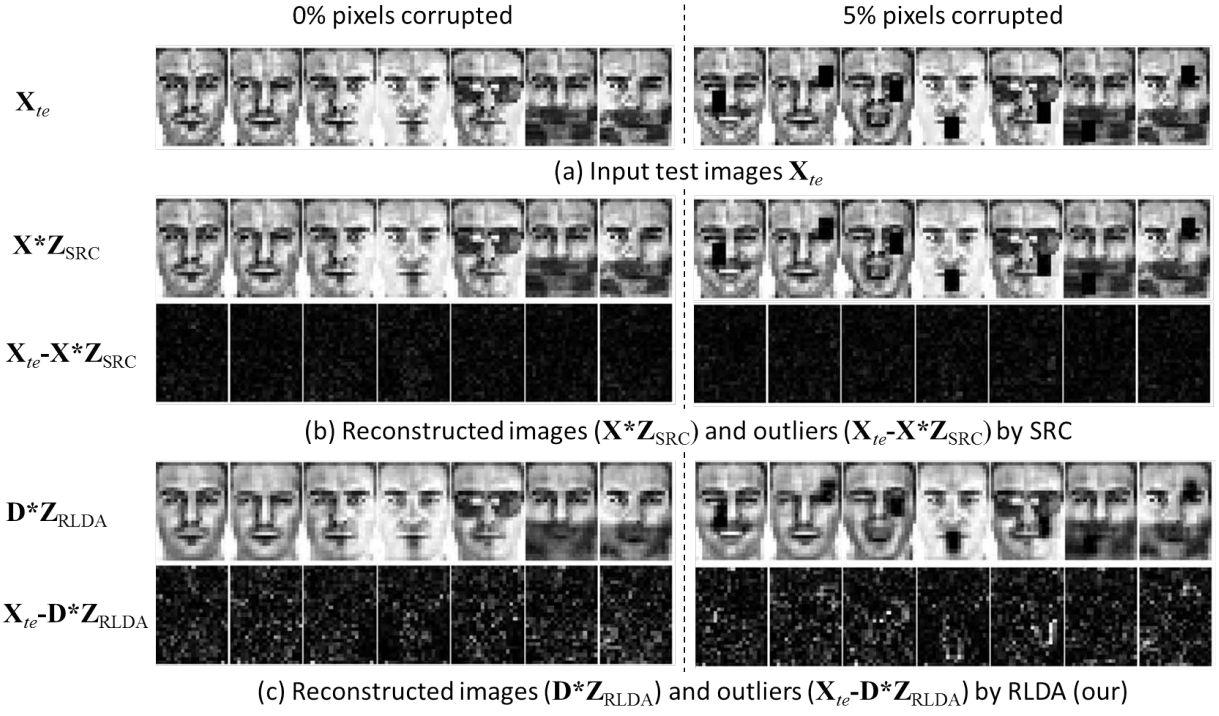


Fig. 5. Decomposition of downsampled test images $\mathbf{X}_{te}$ in the AR face database [41]. **Left**: Experiments on original images (0% corruption). **Right**: Experiments on synthetically corrupted images (5% corruption). (a) Input test images; (b) Reconstructed test images ($\mathbf{X}\mathbf{Z}_{SRC}$) and the outliers ($\mathbf{X}_{te} - \mathbf{X}\mathbf{Z}_{SRC}$) by Sparse Representation for Classification (SRC) [42], where $\mathbf{X}$ is the training images and $\mathbf{Z}_{SRC}$ is the sparse coefficient for the test images $\mathbf{X}_{te}$; (c) Reconstructed test images ($\mathbf{D}\mathbf{Z}_{RLDA}$) and the outliers ($\mathbf{X}_{te} - \mathbf{D}\mathbf{Z}_{RLDA}$) by Robust LDA (RLDA), where $\mathbf{D}$ is the cleaned training images by solving Eq. 6, and $\mathbf{Z}_{RLDA}$ is the RLDA coefficient computed by Eq. 8. Note that RLDA cleaned more intra-sample outliers and reconstructed more facial details than SRC.

rupted images.

We used the AR database [41], which contains over $4,000$ frontal face images of 126 subjects under illumination change, expressions, and facial disguises. 26 pictures were taken for each subject and organized in two sessions. In the experiment, we used the cropped and aligned face images of 50 male subjects and 50 female subjects provided in [41]. For each subject, 13 images from Session 1 were used for training and the remaining 13 images from Session 2 were used for testing. Each image was cropped and resized to 165×120 and then converted to gray-scale (see the first row on the left of Fig. 5 for examples). To evaluate the robustness of the algorithms, we corrupted the images by adding black squares (see the first row on the right half of Fig. 5 for examples).

We followed the settings in [42] and used the two types of features that produced the highest performance of Sparse Representation for Classification (SRC) in [42]: (1) Downsampled face: downsample the cropped images by $1/6$ and vectorize a downsampled image into a 540 dimensional vector; (2) Laplacian face: compute Laplacian face features [43] on the original 165×120 image and select the top 540 components. Fig. 5 illustrates decomposition of downsampled test images $\mathbf{X}_{te}$ (a) in the AR face database [41] by SRC and our Robust LDA (RLDA) approach. The **Left** part of Fig. 5 shows experiments on original images (0% corruption). The **Right** part of Fig. 5 shows experiments on synthetically corrupted images (5% corruption). Using SRC [42] (Fig. 5(b)), the test images were reconstructed as $\mathbf{X}\mathbf{Z}_{SRC}$, where $\mathbf{X}$ represents the training images and $\mathbf{Z}_{SRC}$ is the sparse coefficient. The outliers were then computed as ($\mathbf{X}_{te} - \mathbf{X}\mathbf{Z}_{SRC}$). Note that SRC produced little outliers. This is because both the training and testing images of the same subject contain similar expression, illumination and accessories such as glasses and scarf. SRC

computed the sparse representation of test images $\mathbf{X}_{te}$ using similar training images in $\mathbf{X}$. Fig. 5(c) shows the reconstructed test images ($\mathbf{DZ}_{RLDA}$) and the outliers ($\mathbf{X}_{te} - \mathbf{DZ}_{RLDA}$) by RLDA, where $\mathbf{D}$ represents the cleaned training images by solving Eq. 6 and $\mathbf{Z}_{RLDA}$ is the RLDA coefficient obtained by Eq. 8. Note in contrast to SRC, our RLDA approach used the cleaned training images $\mathbf{D}$ instead of the original training images $\mathbf{X}$. We can see from Fig. 5(c) that RLDA cleaned more intra-sample outliers and reconstructed more facial details than SRC.

In Table 3, we compared face recognition accuracy of linear SVM, SRC and RLDA using both the downsampled images and the Laplacian face as the classification features. As shown in the first row (0%) in Table 3, RLDA produced higher accuracy than SRC and SVM on downsampled images, and comparable accuracy to SRC on Laplacian features. From the 2^{nd} to 4^{th} row, as corruption increased, all methods showed lower accuracy. Furthermore, because the Laplacian features were not computed in the robust manner, under large percentage of corruption (the 3^{rd} to 4^{th} row in Table 3), the results with Laplancian features were worse than RLDA with the downsampled images. Comparing to SVM and SRC, RLDA showed the best robustness, consistently producing the best results.

TABLE 3
Face recognition accuracy on AR face database [41] under synthetic corruption. The percentages in the brackets denotes the portion of images covered by the synthetic squares. *Higher* value indicates better performance. Best results are in bold.

%-pixel corruption	1-NN	SVM	SRC	RLDA
Downsample (0%)	68.5%	76.4%	88.0%	**89.8%**
Laplacian (0%)	90.8%	80.6%	**94.7%**	**94.8%**
Downsample (5%)	33.7%	64.7%	80.8%	**85.1%**
Laplacian (5%)	54.5%	74.5%	71.7%	**77.2%**
Downsample (20%)	9.7%	44.5%	67.5%	**72.4%**
Laplacian (20%)	47.8%	**67.9%**	63.6%	64.9%
Downsample (40%)	7.4%	35.5%	52.9%	**61.4%**
Laplacian (40%)	33.7%	**56.5%**	48.2%	51.4%

5.2.2 RLDA for real databases

This section evaluates our Robust LDA (RLDA) method on two multi-label and one multi-class classification tasks: object categorization on the MSRC dataset, action recognition in the MediaMill dataset and event video indexing on the TRECVID 2011 dataset. Each dataset's corpus and features are described below:

MSRC Dataset (Multi-label)[3] has 591 photographs (see Fig. 6(a)) distributed among 21 classes with an average of 3 classes per image. We mimic [1] dividing each image into an 8×8 grid and calculating the first and second order moments for each color channel on each grid in the RGB space. This results in a 384 dimensional vector, which we use to describe each image.

Mediamill Dataset (Multi-label) [44] consists of 43907 sub-shots (see Fig. 6(c)) divided into 101 classes. We followed [1] and eliminated classes containing fewer than 1000 samples, leaving 27 classes. Then, we randomly selected 2609 sub-shots such that each class has at least 100 labeled data points. Each image was therefore characterized by a 120-dimensional feature vector, as described in [44].

PASCAL VOC 2007 Dataset (Multi-label) consists of 9963 images labeled with at least one of 20 classes, split into `trainval` and `test` sets. We used state of the art features obtained from Overfeat, a Convolutional Neural Network trained on ImageNet [45]. We rescaled every image to 221×221 pixels and obtained a single 4096 dimensional feature vector as the output from layer 22 of the network for every image in the dataset.

TRECVID 2011 Dataset (Multi-class)[4] consists of video data in MED 2010 and the development data of MED 2011, totaling 9822 video clips belonging exclusively to one of 18 classes. We first detected 100 shots for each video and then used their center frames as keyframes. We described each keyframe using dense SIFT descriptors. From these, we learned a 4096 dimension Bag-of-Words dictionary. Each video was represented by a normalized histogram of all of its feature points. We used a 300 core cluster to extract the SIFT features, which took about 1500 CPU hours in total. In the experiment, we randomly split the dataset into two subsets: 3122 entries for training and 6678 for testing.

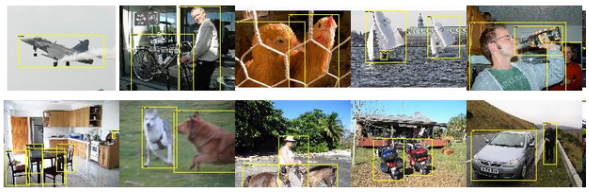
We compared RLDA to the state-of-the-art approach for Multi-Label LDA (MLDA) [1] and to Robust PCA [29] followed by traditional LDA (RPCA+LDA). As a control, we also compared to LDA, PCA+LDA (preserving 99.9% of energy) and a linear one-*vs*.-all SVM.

For the classic LDA-based testing procedure, one first projects the test points using the learned $\mathbf{T}$ from training. Then, for each projected test sample, we find the k-nearest-neighbor (kNN) from the training samples projected by $\mathbf{T}$. Finally, we select the class label from the class labels of k-neighbors by majority voting. However, this procedure is not appropriate in our evaluation for two reasons: (1) it is not fair to use a fixed k for classes with different number of samples, *e.g.*, samples per class are in $[19, 200]$ for MSRC and $[100, 2013]$ for Mediamill; and (2) kNN introduces nonlinearity to the LDA-based classifiers, which is unfair to linear SVM. For these reasons, we use Area Under Receiver Operating Characteristic (AUROC) as our evaluation metric. AUROC summa-

3. http://research.microsoft.com/en-us/projects/ObjectClassRecognition/

4. http://www-nlpir.nist.gov/projects/tv2011/

(a)



(b)

(c)

Fig. 6. Multi-label datasets for object recognition and action classification. Example images in (a) MSRC and (b) PASCAL VOC 2007 , and (c) example keyframes in Mediamill .

rizes the cost/benifit ratio over all possible classification thresholds. We report the average AUROC (over 5-fold Cross Validation) for each method under their best parameters in Table 4. In the MSRC dataset results in Table 4, LDA performs the worst since it is most sensitive to the noise in data. SVM performs better than PCA+LDA and RPCA+LDA. Our method (RLDA) leads to significant improvements over the others due to its joint classification and data cleaning (for both Gaussian and sparse noise in the input). For Mediamill, LDA is just slightly worse than PCA+LDA and RPCA+LDA due to the low noise level in the data. In this case, RLDA does not "over-clean" the data, and performs similar to PCA+LDA and RPCA+LDA. In the PASCAL VOC 2007 dataset results, performance increases become less accentuated, with baseline methods yielding good performance due to the recent advances in representation provided by Overfeat [45]. MLDA, on the other hand, results in a poorer score because it relies heavily on the normalization based on inter-class correlations.

To test our method in a large scale dataset, we ran experiments on the TRECVID2011 dataset. We used the Minimum Normalized Detection Cost (MinNDC), the evaluation criteria for MED 2010 and MED 2011 challenges, as suggested by NIST. Fig. 7 shows that RLDA achieved the best class-wise MinNDC for 9 out of 18 classes over other linear methods, *i.e.*, LDA/MLDA, SVM and RPCA+LDA. Note that because the classes are mutually exclusive, MLDA is identical to LDA. SVM is heavily affected by outliers for the "Wedding Ceremony", "Getting a vehicle unstuck" and "Making a sandwich" cases. For some classes, LDA and RPCA+LDA are similar to or better than RLDA. We believe this is due to (1) the data features computed by Bag-of-Words model smoothed/regularized some outliers, and (2) the nonlinear nature of the classification task. Therefore some error patterns modeled by LDA and RPCA enhanced their discriminative ability. Nevertheless, among all linear algorithms, our method (RLDA) obtained the best average MinNDC. In addition, to show how nonlinearity affects the performances, we compared the kernelized version of the LDA (KLDA), RPCA+LDA (KRPCA+KLDA) and RLDA (KRDA). Here, we apply the homogeneous kernel maps technique [46] to obtain a three order approximation of the χ^2 kernel. Other more accurate approximations are possible [47]. Fig. 7 shows that KRDA still obtains better results. 9 out of 18 best class-wise MinNDC and best average MinNDC over all classes.

TABLE 4
AUROC for Multi-label Object (MSRC), Action (Mediamill) and Image (Pascal VOC) classification. *Higher* value indicates better performance. Best results are in bold.

Database	LDA	SVM	PCA+LDA	MLDA	RPCA+LDA	RLDA
MSRC	0.65	0.79	0.76	0.63	0.75	**0.83**
Mediamill	**0.77**	0.64	**0.77**	0.67	**0.77**	0.76
Pascal VOC2007	0.92	0.90	0.92	0.79	0.87	**0.94**

5.3 RLDA with missing data

This section illustrates the use of RLDA-Missing to perform attribute classification on the PubFig [48] and Multi-PIE [40] databases. Our goal is to show that RLDA-missing can incorporate information from feature vectors with different dimensionality, 49 landmarks on the PubFig images and 66 landmarks on Multi-PIE images.

The PubFig database [48] consists of 58,797 images of 200 people collected from the internet. Classifiers will be trained to recognize the facial attributes, e.g., Gender, race, and accessaries, from image features. The images in the PubFig database were taken in completely uncontrolled situations with non-cooperative subjects. Thus, there are large variations in pose, lighting, expression, occlusion, scene and camera parameters. These imaging conditions pose great difficulties in classifying facial attributes. In addition to those from the PubFig database, we also

Events \ Methods	LDA/MLDA	SVM	RPCA+LDA	RLDA	KLDA	KRPCA+KLDA	KRDA
Maling a cake	1.003	1.004	0.999	**0.937**	0.982	0.972	**0.891**
Batting a run	0.699	1.002	0.950	**0.563**	0.741	0.993	**0.731**
Assembling a shelter	0.999	1.015	1.003	**0.967**	1.004	**0.938**	**0.936**
Attempting a board trick	1.002	1.002	1.006	**0.958**	0.949	**0.882**	0.902
Feeding an animal	1.004	**0.990**	1.004	1.134	**0.989**	1.008	1.084
Landing a fish	0.960	1.002	0.917	**0.856**	0.894	0.994	**0.867**
Wedding ceremony	0.997	12.450	0.979	**0.941**	0.805	0.979	**0.636**
Working on a woodworking project	1.005	**0.859**	1.004	0.900	1.003	**0.816**	0.867
Birthday party	0.986	0.956	**0.937**	1.096	**0.965**	0.985	1.007
Changing a vehicle tire	0.986	**0.984**	1.002	1.011	**0.923**	0.986	0.982
Flash mob gathering	0.838	0.968	0.893	**0.827**	**0.791**	0.994	**0.797**
Getting a vehicle unstuck	0.985	11.672	0.966	**0.907**	**0.952**	1.008	1.002
Grooming an animal	**0.969**	1.002	0.987	0.983	0.992	**0.896**	1.003
Making a sandwich	**1.002**	4.058	1.013	1.042	0.994	**0.966**	1.006
Parade	0.993	**0.972**	0.993	1.043	1.001	**0.988**	1.015
Parkour	0.984	1.002	**0.934**	0.995	0.841	1.006	**0.982**
Repairing an appliance	0.937	**0.600**	1.008	0.823	0.931	0.870	**0.823**
Workin on a sewing project	1.006	1.006	1.003	**0.948**	0.935	1.002	**0.917**
Average Score	0.964	2.363	0.978	**0.941**	0.927	0.960	**0.914**

Fig. 7. MinNDC results for Media Event Detection on TREC2011. *Lower* value indicates better performance. Best results are in bold.

used 5683 face images from the Multi-PIE database from 249 subjects. In the Multi-PIE database, Each subject posed 2 11 facial expressions. For each facial expression, we used the photos taken under frontal lighting and seven horizontal head pose angles $\{-45°, -30°, 15°, 0°, +15°, +30°, +45°\}$.

Our goal is to predict 7 facial attributes (Gender, Asian, White, Indian, Black, Glasses and Beard/Mustache) from facial features. We formulated the facial attribute recognition as a multi-label classification problem: for each image, we assigned 7 attributes that are represented with a binary indicator vector $\mathbf{y}_i \in \Re^{7\times 1}$, where $y_{ij} = 1$ if $\mathbf{x}_i$ belongs has attribute j and $y_{ij} = 0$ ($j = 1, \cdots, 7$) otherwise. To train our facial attribute detector, we used training images from the PubFig database, which have been labeled with 49 landmarks using the supervised descent method [49], and images from Multi-PIE database [40], which have been manually labeled with 68 landmark points. Learning a classifier using both datasets is a challenging problem because the regressor will have input features of different dimensions. In this section, we will show how RR is able to merge information from these two databases to get improved results on estimating facial attributes. During testing (see section 3.2), a test data sample $\mathbf{x}_{te}$ is cleaned to produce $\mathbf{d}_{te}$ and the indicator vector $\mathbf{y}_{te} = \mathbf{T}[\mathbf{d}_{te}; 1] \in \Re^7$. $\mathbf{y}_{te}$ is used as decision values to compute AUROC, or to produce binary class labels using the k-nearest-neighbor method.

Given the images that have been labeled with the seven attributes, we computed the image features as follows. Given the landmarks, we computed an 8-dimensional Histogram of Gradient (HoG) vector around each facial point, (the size of each pixel block is $1/6$ of the length of the nose). Then, we concatenated all the HoG values to form an $8 \times 49 = 392$-dimensional feature vector for the image. See Fig. 8 for an example. In the case of the Multi-PIE images the faces had been manually labeled with 66 landmarks and we proceeded as before, extracting a $8 \times 66 = 544$ dimensional feature vector, see Fig. 8 (b).

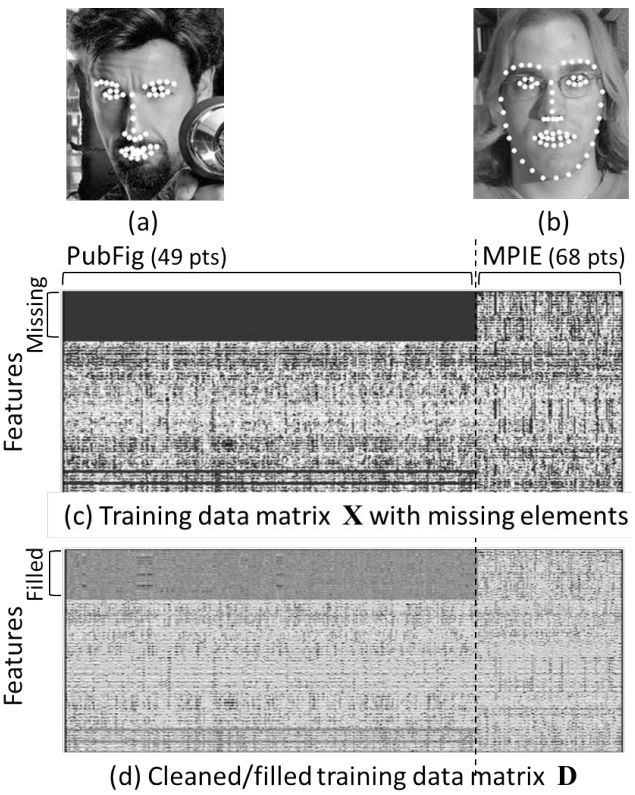


Fig. 8. Training RLDA-Missing classifier on a concatenated data matrix $\mathbf{X}$ consisting of data from the PubFig database (49 facial points detected) (a) and the MultiPIE database (68 facial points detected) (b). In the original concatenated matrix "$\mathbf{X}$" (c), note that the data block of PubFig contains missing elements. In the clean/filled data matrix "$\mathbf{D}$" (d), the missing elements are automatically filled. In testing, we only use the PubFig part of $\mathbf{D}$ to clean the testing data.

We ran four experiments. The first one was a baseline experiment using only the 58797 images from PubFig database labeled with 49 landmarks (392-dimensional feature vectors), we refer to this experiment as "PubFig (49 pts) only". The second experiment, we added the 5683 images from the Multi-PIE database, but we only used 49 of the 68 available landmarks, that are common to both databases. We refer to this experiment as "PubFig (49 pts)&MultiPIE (49 pts)". All feature vectors in the second experiment have 392 dimensions. In the third experiment, we added the same 5683 images from the Multi-PIE database but included all 68 landmarks that are available. The $544-392=52$-dimensional unavailable features in the PubFig dataset are considered as missing data, see Fig. 8 (c) for the concatenated training data matrix "$\mathbf{X}$". We trained RLDA with missing data as described in Section 4, the missing elements in "$\mathbf{X}$" were filled in the cleaned/filled training data the "$\mathbf{D}$" (Fig. 8 (d)). We refer to this experiment as "PubFig (49 pts)&MultiPIE (68 pts)". Finally, we compared RLDA missing with LDA-missing [50], a LDA-based approach for missing data that does not incorporate robustness into their formulation.

In all experiments, we performed grid-search for RLDA parameters (η and λ) with a 4-fold cross-validation. At each trial of cross-validation, we used three PubFig folds and all Multi-PIE images for training, leaving one PubFig fold out for testing. Compared to the two baseline methods ("RLDA: PubFig only" and "RLDA: PubFig (49pts)&MPIE (49pts)"), our RLDA-missing approach can incorporate additional 52-dimensional features from the Multi-PIE dataset. This typically leads to improved classification results. Compared to "LDA-missing" [50], our approach does not rely on explicit assumption on the missing values and adds robustness. "LDA-missing" [50] explicitly models the missing values by Gaussian distribution, whereas the missing elements in this experiment were structured (blocked). As shown in Table. 5, our RLDA-missing produced improved results in both class-wise and average AUROCs.

6 Conclusion

This paper addressed the problem of robust discriminative learning and presented a convex formulation for RR. Our approach jointly learns a regression while removing the outliers that are not correlated with labels or regression outputs. The framework of RR is useful to solve problems such as robust LDA, multi-labeled image classification and regression with missing data. We illustrated the benefits of RR in several computer vision problems including facial attribute detection, head pose estimation, and image/video classification. We show that by removing outliers, our methods consistently learn better representations and outperform state-of-the-art methods in both the linear and kernel spaces (using homogeneous kernel maps). Finally, our approach is general and can easily be applied to make other subspace methods, such as partial least square or canonical correlation analysis, more robust.

Acknowledgments

The second author was supported by the Portuguese Foundation for Science and Technology through the CMU-Portugal program under the project FCT/CMU/P11. The authors would like to thank Francisco Vicente for the assistance with the experiment on the TRECVID 2011 Dataset.

References

[1] H. Wang, C. Ding, and H. Huang, "Multi-label linear discriminant analysis," in *ECCV*, 2010.
[2] D. Huang, M. Storer, F. De la Torre, and H. Bischof, "Supervised local subspace learning for continuous head pose estimation," in *CVPR*, 2011.
[3] R. Plackett, "Some theorems in least squares," *Biometrika*, vol. 37, no. 1-2, pp. 149–157, 1950.
[4] P. Huber, *Robust Statistics*. Wiley and Sons, 1981.
[5] P. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection*. Wiley, 2003.
[6] P. Meer, *Robust Techniques for computer vision, Book chapter in Emerging Topics in Computer Vision, G. Medioni and S. Kang (Eds.)*. Prentice Hall, 2004.
[7] P. Rousseeuw, "Least median of squares regression," *Journal of the American Statistical Association*, vol. 79, no. 388, pp. 871–880, 1984.
[8] J. Gillard, *An Historical Overview of Linear Regression with Errors in both variables*. Cardiff University, School of Mathematics, Technical Report, 2006.
[9] S. Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*. SIAM, 1991.
[10] D. Lindley, "Regression lines and the linear functional relationship," *Journal of the Royal Statistical Society - Supplement*, vol. 9, pp. 218–244, 1947.
[11] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[12] P. Torr and A. Zisserman, "Mlesac: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, pp. 138–156, 2000.
[13] S. Choi, T. Kim, and W. Yu, "Performance Evaluation of RANSAC Family," in *BMVC*, 2009.
[14] R. Adcock, "A problem in least squares," *Analyst*, vol. 5, no. 2, pp. 53–54, 1878.
[15] C. Kummel, "Reduction of observed equations which contain more than one observed quantity," *Analyst*, vol. 6, pp. 97–105, 1879.
[16] A. Wald, "The fitting of straight lines if both variables are subject to error," *Annals of Mathematical Statistics*, vol. 11, pp. 285–300, 1940.
[17] J. Gillard and T. Iles, *Method of moments estimation in linear regression with errors in both variables*. Cardiff University, School of Mathematics, TR, 2005.
[18] B. Matei and P. Meer, "Estimation of nonlinear errors-in-variables models for computer vision applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1537–1552, 2006.
[19] C. Chork and P. Rousseeuw, "Integrating a high-breakdown option into discriminant analysis in exploration geochemistry," *Journal of Geochemical Exploration*, vol. 43, pp. 191–203, 1992.
[20] D. Hawkins and G. McLachlan, "High-breakdown linear discriminant analysis," *Journal of the American Statistical Association*, vol. 92, pp. 136–143, 1997.

TABLE 5
AUROCs of facial attribute classification on the PubFig data. Each row contains results using different method and training data, as specified in the first column "Methods: training data". *Higher* value indicates better performance. Best results are in bold.

Methods: training data \ Attributes	Gender	Asian	White	Indian	Black	Glasses	Beard	Average over classes
RLDA: PubFig (49pts) only	**0.92**	0.50	0.60	0.50	**0.72**	0.77	0.68	0.67
RLDA:PubFig (49pts)&MPIE (49pts)	0.90	0.62	0.57	**0.60**	0.69	0.76	0.70	0.69
LDA-missing [50]: PubFig (49pts)&MPIE(68pts)	0.82	0.57	0.54	0.59	0.61	0.78	0.67	0.65
RLDA-missing: PubFig (49pts)&MPIE(68pts)	0.91	**0.66**	**0.70**	0.56	0.69	**0.81**	**0.71**	**0.72**

[21] X. He and W. Fung, "High breakdown estimation for multiple populations with applications to discriminant analysis," *Journal of Multivariate Analysis*, vol. 72, pp. 151–162, 2000.

[22] C. Croux and C. Dehon, "Robust linear discriminant analysis using s-estimators," *Canadian Journal of Statistics*, vol. 29, 2001.

[23] S. Kim, A. Magnani, and S. Boyd, "Robust FDA," in *NIPS*, 2005.

[24] Y. Zhang and D.-Y. Yeung, "Worst-case linear discriminant analysis," in *NIPS*, 2010.

[25] S. Fidler, D. Skocaj, and A. Leonardis, "Combining reconstructive and discriminative subspace methods for robust classification and regression by subsampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 337–350, 2006.

[26] A. Leonardis and H. Bischof, "Robust recognition using eigenimages," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 99–118, 2000.

[27] H. Jia and A. Martinez, "Support vector machines in face recognition with occlusions," in *CVPR*, 2009.

[28] F. De la Torre and M. Black, "A framework for robust subspace learning," *International Journal of Computer Vision*, vol. 54, pp. 117–142, 2003.

[29] E. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM*, vol. 58, no. 3, 2011.

[30] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *NIPS*, 2009.

[31] Q. Ke and T. Kanade, "Robust l1-norm factorization in the presence of outliers and missing data," in *CVPR*, 2005.

[32] R. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino, "Matrix completion for multi-label image classification," in *NIPS*, 2011.

[33] Z. Zhang, X. Liang, and Y. Ma, "Unwrapping low-rank textures on generalized cylindrical surfaces," in *ICCV*, 2011.

[34] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan, "Multi-task low-rank affinity pursuit for image segmentation," in *ICCV*, 2011.

[35] F. De la Torre, "A least-squares framework for component analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 6, pp. 1041–1055, 2012.

[36] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[37] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low rank representation," in *NIPS*, 2011.

[38] 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, 2007.

[39] G. Golub and C. V. Loan, "Regression lines and the linear functional relationship," *SIAM Journal on Numerical Analysis*, vol. 17, no. 6, pp. 883–893, 1980.

[40] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "The CMU multi-pose, illumination, and expression (multi-pie) face database," CMU Robotics Institute.TR-07-08, Tech. Rep., 2007.

[41] A. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228–233, 2001.

[42] J. Wright, A. Yang, A. Ganesh, 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.

[43] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, 2005.

[44] C. Snoek, M. Worring, J. Gemert, J.-M. Geusebroek, and A. Smeulders, "The challenge problem for automated detection of 101 semantic concepts in multimedia," in *ACM MM*, 2006.

[45] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *ICLR*, 2014.

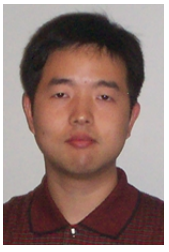
[46] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 480–492, 2012.

[47] F. Li, G. Lebanon, and C. Sminchisescu, "Chebyshev Approximations to the Histogram χ^2 Kernel," in *CVPR*, 2012.

[48] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar, "Attribute and simile classifiers for face verification," in *ICCV*, Oct 2009.

[49] X. Xiong and F. De la Torre, "Supervised descent method and its application to face alignment," in *CVPR*, 2013.

[50] B. Marlin, *Missing Data Problems in Machine Learning*. PhD thesis, University of Toronto, 2008.



Dong Huang received his M.Sc. in Automation and PhD degrees in Computer Science from University of Electronic Science and Technology of China, respectively, in 2005 and 2009, Chengdu, China. In 2009 and 2012 he became postdoctoral research associate and project scientist the Robotics Institute at Carnegie Mellon University, USA. He is a member of the Human Sensing Lab (http://humansensing.cs.cmu.edu/). His research focuses on computer vision (facial and human actions) and machine learning (parameterized regression).

Ricardo S. Cabral is a PhD student from Carnegie Mellon and IST-Lisbon. He received his Masters in Electrical and Computer Eng. at IST-Lisbon in 2009. He received an outstanding academic achievement award in 2008 from IST-Lisbon, where he worked in several projects, including video handling for the 2012 London Olympics. His research focuses on low rank models for computer vision and machine learning.

Fernando De la Torre received his B.Sc. degree in Telecommunications, M.Sc. and Ph. D degrees in Electronic Engineering, respectively, in 1994, 1996 and 2002, from La Salle School of Engineering in Ramon Llull University, Barcelona, Spain. In 1997 and 2000 he became Assistant and Associate Professor in the Department of Communications and Signal Theory in La Salle School of Engineering. In 2003 he joined the Robotics Institute at Carnegie Mellon University and he is currently Associate Research Professor.

His research interests are in the fields of Computer Vision and Machine Learning. Currently, he is directing the component analysis lab (http://ca.cs.cmu.edu) and co-directing the human sensing lab (http://humansensing.cs.cmu.edu). Dr. De la Torre has co-organized the first workshop on component analysis methods for modeling, classification and clustering problems in computer vision in conjunction with CVPR'07 and the workshop on human sensing from video in conjunction with CVPR'06. He has also given several tutorials at international conferences on the use and extensions of component analysis methods.