Robust Regression

Dong Huang, Ricardo Cabral and Fernando De la Torre

Robotics Institute, Carnegie Mellon University

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. Regression methods typically map image features ($\mathbf{X}$) to continuous (*e.g.*, pose) or discrete (*e.g.*, object category) values. A major drawback of existing regression methods is that samples are directly projected onto a subspace and hence fail to account for outliers which are common in realistic training sets due to occlusion, specular reflections or noise. It is important to notice that in existing regression methods, and discriminative methods in general, the regressor variables $\mathbf{X}$ are assumed to be noise free. Due to this assumption, discriminative methods experience significant degrades in performance 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, multi-label classification and head pose estimation from images. Several synthetic and real world examples are used to illustrate the benefits of RR.

Key words: Robust methods, errors in variables, intra-sample outliers

1 Introduction

Discriminative methods (*e.g.*, kernel regression, SVM) have been successfully applied to many computer vision problems. Unlike generative approaches that produce a probability density over all variables, discriminative approaches provide a direct 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 is available.

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

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

[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

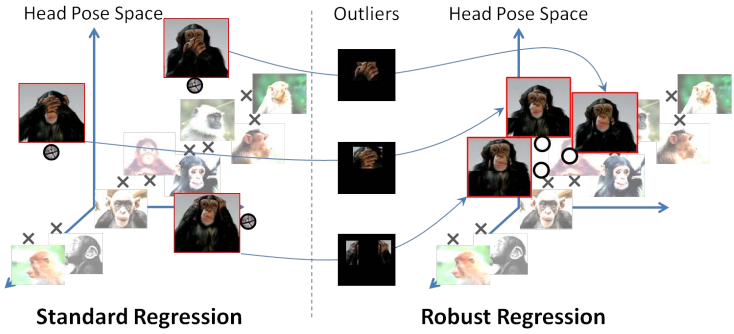


Fig. 1: The goal is to predict the yaw angle of the monkey head from image features. Note the image features (image) contains outliers (hands of the monkey). (Left) Standard regression: projects the partially occluded frontal face images *directly* onto the head pose subspace and fails to estimate the correct pose; (Right) Robust regression removes the intra-sample outliers and projects only the cleaned input images without biasing the pose estimation.

In the training stage, we learn the mapping $\mathbf{T}$, and in testing we estimate the pose by projecting the image features of the test image, $\mathbf{T}\mathbf{x}_{test}$. It is important to notice that in training and testing, we assume $\mathbf{X}$ to be noise free. A single outlier can bias the projection because we project the data *directly* onto the subspace of $\mathbf{T}$. For instance, $\mathbf{T}\mathbf{x}_{test}$, the dot product of $\mathbf{x}_{test}$ with each row of $\mathbf{T}$, can be largely biased by only one outlier. For this reason, existing discriminative methods lack robustness to outliers.

Standard regression, Eq. (1), is optimal under the assumption that the error, $\mathbf{E} = \mathbf{Y} - \mathbf{T}\mathbf{X}$, is normally distributed. However, it is well known that a small number of gross outliers can arbitrarily bias the estimation of the model's parameters. This is a thoroughly studied problem in statistics, and the last decades have witnessed the fast paced development of the so-called robust methods [3–5]. However, all these traditional robust approaches for regression are different from the problem addressed in this paper. There are two main differences: (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 (the whole image is an outlier). This work proposes an intra-sample RR method that explicitly accounts for outliers in $\mathbf{X}$. Our work is related to errors in variables (EIV) models (*e.g.*, [6–8]). However, unlike existing EIV models, RR does not need to have a prior estimate of the noise and all parameters are automatically estimated. We illustrate the power of RR in several computer vision tasks including head pose estimation from images and robust lda for multi-label image classification.

represent scalar variables. $\|\mathbf{A}\|_F^2$ designates the Frobenius norm of matrix $\mathbf{A}$. $\|\mathbf{A}\|_*$ is the Nuclear Norm (sum of singular values) of $\mathbf{A}$. ℓ_0 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 1s. $\mathbf{0}_{k\times n} \in \Re^{k\times n}$ 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.

2 Related Work

There exist extensive literature on robust methods for regression. Huber [3] introduced M-estimation for regression, providing robustness to sample outliers. Rousseeuw and Leroy proposed Least Trimmed Squares [4], 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 [9] (and its ML and M-estimator variants [10, 11]) randomly subsample input data to construct a tentative model. Model parameters are updated when a new configuration produces smaller inlier error than its predecessors. However, these methods rely on the assumption that the computation of the model parameters of a subset is inexpensive and can only remove sample outliers.

To deal with noise in the variables, Error-In-Variable (EIV) approaches [7] were proposed. 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 [12] or their ratio is known [13]. Under these assumptions, orthogonal regression can minimize the gaussian error orthogonal to the learned regression vectors. Grouping-based methods [14] assume that errors are respectively i.i.d. among the input and respond variables, so that one can split the data into groups and suppress the errors by computing difference of the group sum, geometric means or instrument variables. Moment-based methods [15] learn the regression by estimating the high-order statistics, *i.e.*, moments, from the data of i.i.d. likelihood-based methods [8] learn a reliable regression when the input and respond variables follow a joint, normal and identical distribution. Total Least Square (TLS) [7] and its nonlinear generalization [16], solve for additive/multiple terms that enforce the correlation between the input and respond variables. TLS-based methods relax the assumption in previous methods to allow correlated and non-identical 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.

Independent of the work on EIV for regression, several authors have addressed the issue of robust classification. On one hand, several authors have proposed robust extensions of LDA, where the empirical estimation of the class mean vectors and covariance matrices are replaced by their robust counterparts (*e.g.*, [17]). In machine learning, several authors [18, 19] 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. However, these methods are only robust to sample-outliers.

Our work is more related to recent work in computer vision. Fidler and Leonardis [20] robustify LDA for intra-sample outliers. In the training stage, [20] computed PCA on the training data, replaced the minor PCA components by a robustly estimated basis, and combined the two basis into a new one. Then the data is projected into the combined basis and LDA is computed. During testing, [20] first estimates the coefficients of a test data on the recombined

basis by sub-sampling the data elements using [21]. Finally, the class label of the test data is determined by applying learned LDA on the estimated coefficients. Although outliers outside of the PCA subspace can be suppressed, [20] do not address the problem of learning LDA with outliers in the PCA subspace of the training data. Zhu and Martinez [22] proposed learning a SVM with missing data and robust to outliers. However, [22] requires that 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) no 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 [23] and its recent advances due to rank minimization procedures [24, 25]. These methods model data as the sum of a low-rank clean data component with an arbitrary large and sparse outlier matrix. De La Torre and Black [23] 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). [24, 25] separated a low-rank data matrix from a sparse corruption, despite its arbitrarily large magnitude and unknown pattern. A major theoretical advantage of this approach is the convex formulation. This approach has been extended to other problems such as background modeling and shadow removal [25], image tagging and segmentation [26], texture unwrapping [27] or segmentation [28]. 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)

Let $\mathbf{X} \in \Re^{d \times n}$ be a matrix containing n d-dimensional samples possibly corrupted by outliers. Formally, $\mathbf{X} = \mathbf{D} + \mathbf{E}$, where $\mathbf{D}$ is the underlying noise-free component and $\mathbf{E}$ contains the 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 or 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 $\mathbf{D}$ plus $\mathbf{E}$, and only computing $\mathbf{T}$ using the clean free data $\mathbf{D}$. RR solves the following optimization problem

$$\min_{\mathbf{T},\mathbf{D},\mathbf{E}} \frac{\eta}{2} \|\mathbf{W}(\mathbf{Y} - \mathbf{TDH})\|_F^2 + \text{rank}(\mathbf{D}) + \lambda \|\mathbf{E}\|_0 \quad s.t. \quad \mathbf{X} = \mathbf{D} + \mathbf{E}, \tag{2}$$

where $\mathbf{W} \in \Re^{d_y \times d_y}$ weights the output dimensions, $\mathbf{T}$ is the regression matrix and $\mathbf{H} = \left(\mathbf{I}_n - \mathbf{1}\mathbf{1}^T/n\right)$ is a centering matrix. RR explicitly avoids projecting the outlier matrix $\mathbf{E}$ to the output space by learning the regression $\mathbf{T}$ only from the centered noise-free data $\mathbf{DH}$. The second and third terms of (2) are similar to RPCA [25] in that they respectively constrain $\mathbf{D}$ to a low dimensional subspace

and encourages $\mathbf{E}$ to be sparse. RR is different from RPCA plus regression since it decomposes the input data $\mathbf{X} = \mathbf{D} + \mathbf{E}$ in a supervised manner; that is, the clean data $\mathbf{D}$ will preserve the subspace of $\mathbf{X}$ that correlates 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 Section 4.1).

The original form of RR, Eq. (2), is cumbersome to solve as the rank and cardinality operators are neither convex nor differentiable. Following the techniques in [25], these operators are respectively relaxed to their convex envelopes: the nuclear norm and the ℓ_1-norm. The cost function (2) is rewritten as

$$\min_{\mathbf{T},\mathbf{D},\mathbf{E}} \frac{\eta}{2} \|\mathbf{W}(\mathbf{Y} - \mathbf{TDH})\|_F^2 + \|\mathbf{D}\|_* + \lambda\|\mathbf{E}\|_1 \quad s.t. \quad \mathbf{X} = \mathbf{D} + \mathbf{E},$$

which can be efficiently optimized using an Augmented Lagrange Muliplier (ALM) technique. Let $\hat{\mathbf{D}} = \mathbf{DH}$, we rewrite (3) as

$$\begin{aligned}\min_{\mathbf{T},\mathbf{D},\hat{\mathbf{D}},\mathbf{E}} \; & \frac{\eta}{2} \left\|\mathbf{W}(\mathbf{Y} - \mathbf{T}\hat{\mathbf{D}})\right\|_F^2 + \|\mathbf{D}\|_* + \lambda\|\mathbf{E}\|_1 + \langle \Gamma_1, \mathbf{X} - \mathbf{D} - \mathbf{E}\rangle \\ & + \frac{\mu_1}{2}\|\mathbf{X} - \mathbf{D} - \mathbf{E}\|_F^2 + \langle \Gamma_2, \hat{\mathbf{D}} - \mathbf{DH}\rangle + \frac{\mu_2}{2}\|\hat{\mathbf{D}} - \mathbf{DH}\|_F^2, \end{aligned} \tag{3}$$

where $\Gamma_1 \in \Re^{d\times n}$ and $\Gamma_2 \in \Re^{d\times n}$ are Lagrange multiplier matrices, and μ_1 and μ_2 are the penalty parameters. The resulting algorithm is summarized in Alg .1.

Algorithm 1 ALM algorithm for solving RR (3)

Input: $\mathbf{X}$, $\mathbf{Y}$, parameters η, λ, ρ, γ

Initialization: $\mathbf{T}^{(0)} = \mathbf{YX}^T(\mathbf{XX}^\mathbf{T} + \gamma\mathbf{I}_{d_x})$,$\hat{\mathbf{D}}^{(0)} = \mathbf{T}^{(0)}+\mathbf{Y}$, $\mathbf{E}^{(0)} = \mathbf{X} - \mathbf{D}^{(0)}$,

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 $\frac{\left\|\mathbf{X}-\mathbf{D}^{(k)}-\mathbf{E}^{(k)}\right\|_F}{\|\mathbf{X}\|_F} > 10^{-8}$, $\frac{\left\|\hat{\mathbf{D}}^{(k)}-\mathbf{D}^{(k)}\mathbf{H}\right\|_F}{\left\|\hat{\mathbf{D}}^{(k)}\right\|_F} > 10^{-8}$ **do**

- Update $\mathbf{T}$ (assuming $\mathbf{W} = diag\{w_{ii}\})$:
  $\mathbf{T}^{(k+1)} = [\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_c]$, where $\mathbf{t}_i^\top = w_{ii}^2\mathbf{y}_i\hat{\mathbf{D}}^T(w_{ii}^2\hat{\mathbf{D}}^{(k+1)}(\hat{\mathbf{D}}^{(k+1)})^T + \gamma\mathbf{I}_d)^{-1}$
  and γ regularizes the scale of $\mathbf{t}_i$.
- Update $\hat{\mathbf{D}}$: $\hat{\mathbf{D}}^{(k+1)} = \left[\eta(\mathbf{T}^{(k)})^T\mathbf{W}^T\mathbf{WT}^{(k)} + \mu_2^{(k)}\mathbf{I}_d\right]^{-1}\left[\eta(\mathbf{T}^{(k)})^T\mathbf{WY} - \Gamma_2^{(k)} + \mu_2^{(k)}\mathbf{D}^{(k)}\mathbf{H}\right]$;
- Update $\mathbf{D}$: $\mathbf{D}^{(k+1)} = \mathcal{D}_{1/\beta}(\mathbf{Z}^{(k+1)})$, $\mathbf{Z}^{(k+1)} = \mathbf{D}^{(k+1)} - \frac{1}{\beta}\Big(-\Gamma_1^{(k)} + \mu_1^{(k)}\left[\mathbf{D}^{(k)} - (\mathbf{X} - \mathbf{E}^{(k)})\right] - \Gamma_2^{(k)}\mathbf{H}^T + \mu_2^{(k)}\left[\mathbf{D}^{(k)}\mathbf{H}^{(k)} - \hat{\mathbf{D}}^{(k)}\right]\mathbf{H}^\mathbf{T}$, and $\beta \geq \|\mu_1^{(k)}\mathbf{I} + \mu_2^{(k)}\mathbf{HH}^T\|_F$;
- Update $\mathbf{E}$: $\mathbf{E}^{(k+1)} = \mathcal{S}_{\lambda/\mu_1^{(k)}}\left(\mathbf{X} - \mathbf{D}^{(k)} + \Gamma_1^{(k)}/\mu_1^{(k)}\right)$;
- Update $\Gamma_1^{(k+1)} = \Gamma_1^{(k)} + \mu_1^{(k+1)}(\mathbf{X} - \mathbf{D}^{(k+1)} - \mathbf{E}^{(k+1)})$, $\Gamma_2^{(k+1)} = \Gamma_2^{(k)} + \mu_2^{(k+1)}(\hat{\mathbf{D}}^{(k+1)} - \mathbf{D}^{(k+1)}\mathbf{H})$, $\mu_1^{(k+1)} = \rho\mu_1^{(k)}$, $\mu_2^{(k+1)} = \rho\mu_2^{(k)}$;

end while

Output: $\mathbf{T}$, $\mathbf{D}$, $\mathbf{E}$

3.1 Robust LDA: an extension of 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 ($n < d$), LDA typically suffers from the small sample size problem. One possible solution for this is formulating LDA as a least-squares (LS) problem [29]. LS-LDA [29] directly maps $\mathbf{X}$ to the class labels represented by an indicator matrix. LS-LDA minimizes

$$\min_{\mathbf{T}} \left\| (\mathbf{Y}\mathbf{Y}^T)^{-1/2}(\mathbf{Y} - \mathbf{T}\mathbf{X}) \right\|_F^2, \tag{4}$$

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 and $y_{ij} = 0$ otherwise. The normalization factor $\mathbf{W} = (\mathbf{Y}\mathbf{Y}^T)^{-1/2}$ compensates for different number of samples per class. $\mathbf{T} \in \Re^{c \times d}$ is a reduced rank regression matrix (which has rank $c-1$ if the data is centered). After $\mathbf{T}$ is learned, a test data $\mathbf{x}_{test} \in \Re^{d \times 1}$ is projected by $\mathbf{T}$ onto the c dimensional output space spanned by $\mathbf{T}\mathbf{X}$, then the class label of the test data $\mathbf{x}_{test}$ is assigned using k-NN.

When $\mathbf{X}$ is corrupted by outliers, Eq. (4) suffers from the same bias problem as standard regression. RR, Eq. (3), can be directly applied to Eq. (4), yielding

$$\min_{\mathbf{T},\mathbf{D},\mathbf{E}} \frac{\eta}{2} \left\| (\mathbf{Y}\mathbf{Y}^T)^{-1/2}(\mathbf{Y} - \mathbf{T}\mathbf{D}\mathbf{H}) \right\|_F^2 + \|\mathbf{D}\|_* + \lambda\|\mathbf{E}\|_1 \quad s.t. \quad \mathbf{X} = \mathbf{D} + \mathbf{E},$$

a Robust LDA formulation which can be easily solved as a special case of RR.

3.2 Testing for new data points

To remove outliers in a new testing sample $\mathbf{X}_t$, we minimize

$$\min_{\mathbf{Q}_t,\mathbf{E}_t} \frac{\eta\|\mathbf{W}\mathbf{T}\|_F^2}{2} \left\| \mathbf{X}_t - (\mathbf{D}\mathbf{1}\mathbf{1}^T/n + \mathbf{U}\mathbf{Q}_t) - \mathbf{E}_t \right\|_F^2 + \lambda\|\mathbf{E}_t\|_1, \tag{5}$$

where $\mathbf{U}$ contains the principal components of the clean data $\mathbf{D}$ (preserving 99.99% energy), $\mathbf{Q}_t$ are the coefficients such that a linear combination of $\mathbf{U}$ can reconstruct the clean part of the data $\mathbf{X}_t$. η and λ are the same parameters used during training. After solving (5), the regression or classification for $\mathbf{X}_t$ is computed as $\mathbf{Y}_t = \mathbf{T}\mathbf{U}\mathbf{Q}_t$.

4 Experimental Results

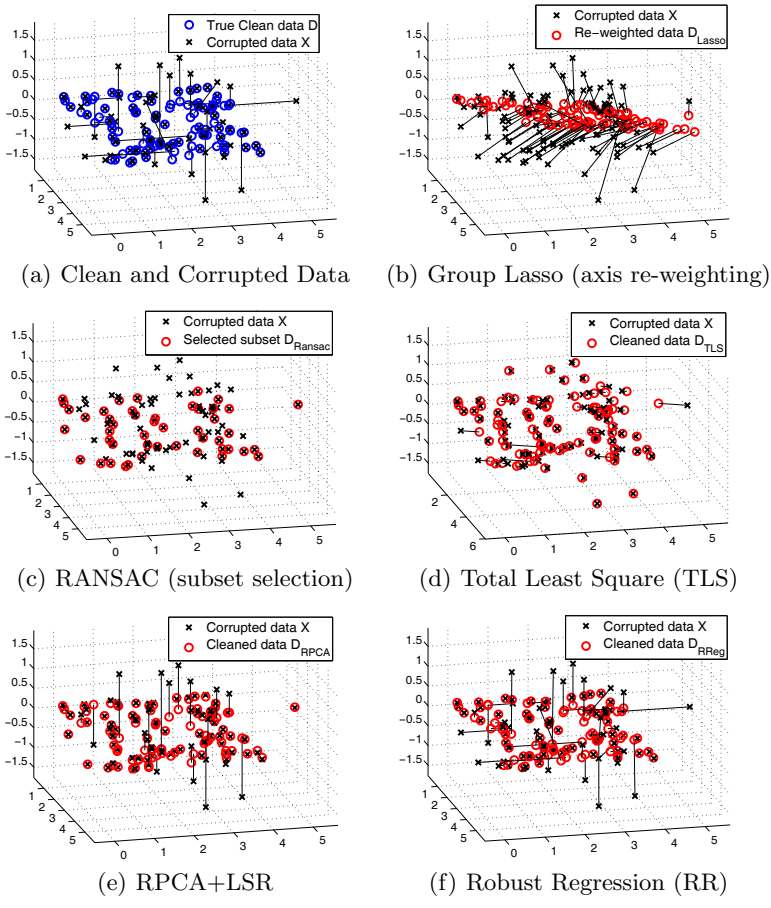
This section compares our RR methods against state-of-the-art approaches on regression and classification. The first experiment uses synthetic data to illustrate the ability of RR to remove in-subspace outliers that existing methods can not detect. The second experiment illustrates the application of RR to the problem of head pose estimation from corrupted images. The final experiments report comparisons of our RR against state-of-the-art multi-label classification algorithms on the MSRC, Mediamill and TRECVID2011 databases.

4.1 Synthetic Data

This section illustrates the benefits of RR in a synthetic example. We have generated 200 three dimensional samples, where the first two components are generated from a uniform distribution between $[0, 6]$, and the third dimension is 0. In Matlab notation, $\mathbf{D} = [6 * rand(2, 200); \mathbf{0}^T]$, $\mathbf{X} = \mathbf{D} + \mathbf{E}$, $\mathbf{Y} = \mathbf{T}_*\mathbf{D}$, where $\mathbf{D} \in \Re^{3\times 200}$ is the clean data. The error term, $\mathbf{E} \in \Re^{3\times 200}$, is 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 ($\sim \mathcal{N}(0, 1)$) in the third dimension, this simulates noise outside the subspace. $\mathbf{T}_* \in \Re^{3\times 3}$ is randomly generated (each element is uniformly sampled in $[0, 1]$), and is used as the true regression matrix. The output data matrix is generated as $\mathbf{Y} = \mathbf{T}_*\mathbf{D} \in \Re^{3\times 200}$. Fig. 2 (a) shows the clean data $\mathbf{D}$ with blue "o"s, and the corrupted data $\mathbf{X}$ with black "×"s. For easiness of visualization, we have only shown 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 $\mathbf{E} = \mathbf{X} - \mathbf{D}$, while the horizontal line segments represent the in-subspace component of $\mathbf{E} = \mathbf{X} - \mathbf{D}$.

We compared our RR with five state-of-the-art methods: (1) Standard least-squares regression (LSR), (2) GroupLasso (GLasso) [30], (3) RANSAC [9], (4) Total Least Square (TLS) [31] that assumes the error in the data is additive and follow a gaussian distribution, (5) RPCA+LSR, which consist on first performing RPCA [24] on the input data, and then learn the regression on the cleaned data using standard LSR. The LSR learns directly 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 select 100 samples for training and 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 within sample outliers. Observe 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 a Gaussian distribution of equal error. Fig. 2(e) shows results obtained by the method RPCA+LSR, that first computes RPCA to clean the data and then LSR. The data cleaned by RPCA [24], $\mathbf{D}_{RPCA}$, 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. Finally, Fig. 2 (f) shows the result of RR. The clean data $\mathbf{D}_{RR}$ is denoted by red"o"s. Observe that our approach is able to clean both the in-subspace outliers (the horizontal lines) and out-of-subspace (the vertical lines). This is because our method computes jointly the regression and the subspace estimation.



(a) Clean and Corrupted Data (b) Group Lasso (axis re-weighting)

(c) RANSAC (subset selection) (d) Total Least Square (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.

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

Table 1: RAE error for $\mathbf{Y}$ and $\mathbf{T}$ for different methods.

	LSR	GLasso	RANSAC	TLS	RPCA+LSR	**RR**
$RAE_{\mathbf{T}}$	0.078	0.078	0.070	0.052	0.074	**0.005**
$RAE_{\mathbf{Y}}$	0.0272	0.0274	0.0263	0.0261	0.0262	**0.011**

4.2 Pose estimation from images

This section illustrates the benefit of RR in the problem of head pose estimation from corrupted images. We used a subset of CMU Multi-PIE database [32] that contains 3707 face images of all 337 subjects from all 4 sessions. For each subject, we used images taken under 11 head poses with yaw angle [$-90°$, $-75°$, $-60°$, $-45°$, $-15°$, $0°$, $15°$, $45°$, $60°$, $75°$, $90°$]. Each image is cropped around the face region and resized to 50×60. We vectorized the images into a vector of 3000 dimensions in the matrix $\mathbf{X} \in \Re^{3000\times 3707}$ and the yaw angles of the images are gathered as the output data $\mathbf{Y} \in \Re^{1\times 3707}$. To evaluate the robustness of the compared methods, we simulate structured occlusions by adding white blocks (0.1 times the image width) at 5 random locations (see Fig. 3a for examples of corrupted images).

Table 2: Yaw angle error for different methods and corruption percentages.

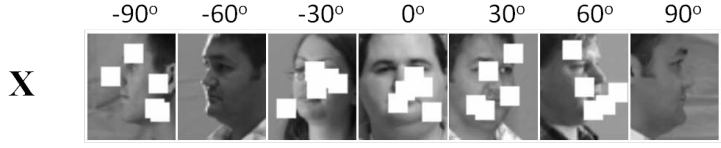
% of corruption	0%	20%	40%	80%
LSR	12.3°	14.5°	15.1°	17.3°
GLasso	16.0°	17.8°	20.2°	21.1°
RANSAC	12.2°	14.1°	14.9°	17.8°
RPCA+LSR	13, 3°	15.4°	18.3°	20.4°
RR	**12.1°**	**13.0°**	**13.7°**	**15.2°**

Similar to the previous section, we have compared RR with four methods to learn a regression from the image $\mathbf{X}$ to the yaw angle $\mathbf{Y}$: (1) LSR, (2) GLasso [30], (3) RANSAC [9], (4) RPCA+LSR. For a fair comparison, we randomly divided the 3707 images into 10 folds and performed 10-fold cross-validation in methods (2)-(4) to compute parameters of interest. The performance of the compared methods is measured with the mean deviations of angle error on all test folders.

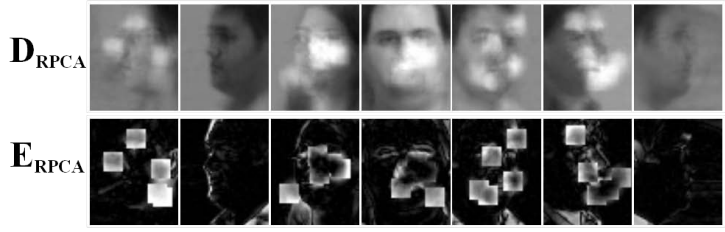
Table 2 summarizes the results of methods (1)-(4) and RR when 0%, 20%, 40%, 80% of the images are corrupted in both the training and testing folders. As expected, the LSR method produced larger angle error with the increasing percentage of outliers. RANSAC produced comparable error as standard LSR indicating 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 lack the ability to preserve the discriminative information in $\mathbf{X}$ that correlates with the angles $\mathbf{Y}$. RR got the smallest error and it is stable w.r.t. the percentage of corruption.

To further illustrate how RR differs from RPCA+LSR, Fig. 3 visualizes the decomposition done by RR, *i.e.*, $\mathbf{X} = \mathbf{D}_{RR} + \mathbf{E}_{RR}$ an by RPCA, *i.e.*, $\mathbf{X} = \mathbf{D}_{RPCA} + \mathbf{E}_{RPCA}$, for the same input images. Images under all pose angles (except $-60°$ and $90°$) are corrupted with white blocks (see Fig. 3(a)). Fig. 3(b)-(c) show that both RPCA and RR are able to remove most of the white blocks. However RR preserves much less personal facial details in $\mathbf{D}_{RR}$ than RPCA in

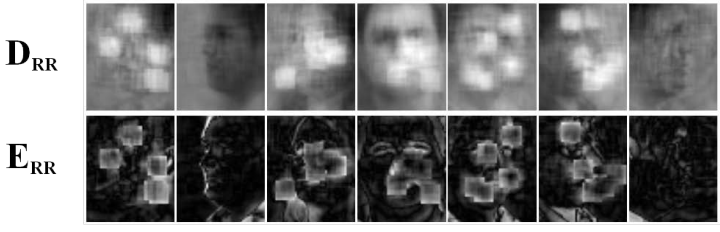
$\mathbf{D}_{RPCA}$ (especially images under pose $-60°$ and $90°$). With less facial details and more dominant profiles, the regression trained on $\mathbf{D}_{RR}$ (as in RR) is able to model higher correlation with the pose angles than using $\mathbf{D}_{RPCA}$. This is why RR tends to be more robust than the RPCA in estimating the pose angles.



(a) Examples of partially corrupted input images $\mathbf{X}$



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



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

Fig. 3: Decomposition of input images in (a) by RPCA (b) and RR (c).

4.3 Robust LDA

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 corpus and features is described below:

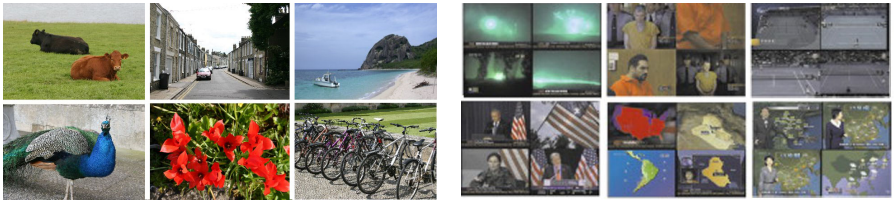
MSRC Dataset (Multi-label)[2] has 591 photographs (see Fig. 4(a)) distributed among 21 classes, with an average of 3 classes per image. We mimic [1] and divide each image into an 8×8 grid and calculate the first and second order moments for each color channel on each grid in the RGB space. This results in a 384 dimensional vector, which we use to describe each image.

Mediamill Dataset (Multi-label) [33] consists of 43907 sub-shots divided in 101 classes. We follow [1] and eliminate classes containing less than 1000 sam-

[2] http://research.microsoft.com/en-us/projects/ObjectClassRecognition/

ples, leaving 27 classes. Then, we randomly select 2609 sub-shots such that each class has at least 100 labeled data points. Each image is therefore characterized by a 120-dimensional feature vector, as described in [33].

TRECVID 2011 Dataset (Multi-class)[3] 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 detect 100 shots for each video and then use their center frames as keyframes. We describe each keyframe using dense SIFT descriptors. From these, we learn a 4096 dimension Bag-of-Words dictionary. Each video is 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 2687 CPU hours in total. In the experiment, we randomly split the dataset into two subsets, with 3122 entries for training and 6678 for testing.



(a) (b)

Fig. 4: Multi-label datasets for object recognition and action classification. Example images in MSRC (a) and example keyframes in Mediamill (b).

We compared RLDA to the state of the art approach for Multi-Label LDA (MLDA) [1], and to Robust PCA [24] followed by traditional LDA (RPCA+LDA). For control, we also compare 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, find k-nearest-neighbor (kNN) from the training samples projected by $\mathbf{T}$; finally 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's 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, $[100, 2013]$ for Mediamill; (2) kNN introduces nonlinearity to the LDA-based classifiers, which is unfair to linear SVM. For these reasons, we use the Area Under Receiver Operating Characteristic (AUROC) as our evaluation metric. AUROC summarizes 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 3. In the MSRC dataset results in Table 3, LDA performs the worst since it's 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

[3] http://www-nlpir.nist.gov/projects/tv2011/

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.

Table 3: AUROC for Multi-label Object (MSRC) and Action (Mediamill) classification. *Higher* value indicates better performance. Best results are in bold.

Database	LDA	SVM	PCA+LDA	MLDA	RPCA+LDA	RLDA
MSRC	0.6463	0.7863	0.7585	0.6313	0.7480	**0.8170**
Mediamill	0.7667	0.6230	0.7702	0.6658	0.7704	**0.7710**

To test our method in a large scale dataset, we run experiments on the TREC2011 dataset. We used the Minimum Normalized Detection Cost (MinNDC), the evaluation criteria for MED 2010 and MED 2011 challenges suggested by NIST. Fig. 5 shows that RLDA achieved the best class-wise MinNDC for 8 out of 18 classes over other linear methods, *i.e.*, LDA/MLDA, SVM and RPCA+LDA. Note for the class-wise cases LDA and MLDA are identical. 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 or better than RLDA. Nevertheless, among all linear algorithms, our method (RLDA) obtains the best average MinNDC. In addition, to show how nonlinearity affects the performances, we compared the kernelized version of the LDA, RPCA+LDA and RLDA. Here we apply the homogeneous kernel maps technique [34] to obtain a three order approximation of the χ^2 kernel. Other approximations are possible [35]. Fig. 5 shows that KRLDA still obtains better results, 13 out of 18 best class-wise MinNDC and best average MinNDC over all classes.

5 Conclusion

This paper addressed the problem of robust discriminative learning, and presents a convex formulation for RR. Our robust approach jointly learns a regression, while removing the outliers that are not correlated with labels or regression outputs. We illustrated the benefits of RR in several computer vision problems ranging from RR for pose estimation, robust LDA to multi-labeled image classification. Experiments 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 be easily applied to robustify other subspace methods such as partial least square or canonical correlation analysis.

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

Event Description \ Methods	LDA/MLDA	SVM	RPCA+LDA	**RLDA**	KLDA	KRPCA+KLDA	**KRLDA**
Maling a cake	1.0027	1.0038	0.999	**0.9091**	0.9819	1.0019	**0.929**
Batting a run	**0.6987**	1.0019	0.9498	0.8552	0.7413	0.929	**0.6832**
Assembling a shelter	0.9989	1.0152	1.0026	**0.9787**	1.0038	1.0019	**0.9744**
Attempting a board trick	**1.0019**	**1.0018**	1.0057	**1.0019**	**0.9494**	0.979	0.9513
Feeding an animal	1.0038	**0.9899**	1.0038	1.0019	**0.9889**	0.995	0.9992
Landing a fish	0.9605	1.0019	0.9169	**0.9056**	0.8937	0.9399	**0.872**
Wedding ceremony	0.9967	12.4498	**0.9789**	0.9923	0.8048	0.8741	**0.7675**
Woodworking project	1.0051	**0.8588**	1.0038	1.0057	1.0032	1.0038	**0.9975**
Birthday party	0.9862	0.9561	**0.9368**	0.9881	0.9654	0.9695	**0.9595**
Changing a vehicle tire	0.9856	0.9842	1.0019	**0.9549**	**0.923**	0.9572	**0.923**
Flash mob gathering	0.8384	0.9675	0.8933	**0.8189**	0.7905	0.7786	**0.734**
Getting a vehicle unstuck	0.9848	11.6719	**0.9659**	0.9867	0.9524	0.9581	**0.9468**
Grooming an animal	**0.9691**	1.0019	0.9868	1.0094	**0.9918**	1.0019	1.0006
Making a sandwich	1.0019	4.0583	1.0132	**0.981**	0.9936	**0.9917**	**0.9917**
Parade	0.9931	**0.9723**	0.9931	0.9805	1.0006	**0.9949**	0.9987
Parkour	0.9837	1.0019	**0.9336**	1.0019	0.8412	0.8211	**0.8203**
Repairing an appliance	0.9369	**0.5998**	1.0075	0.9344	**0.9312**	0.9652	0.9419
Working on a sewing project	1.0057	1.0056	1.0025	**0.9192**	0.9349	0.9544	**0.9054**
Average Score	0.9641	2.3635	0.9775	**0.9571**	0.9273	0.951	**0.9109**

Fig. 5: MinNDC results for Media Event Detection on TREC2011. *Lower* value indicates better performance. Best results are in bold.

authors would like to thank Francisco Vicente for the assistance with the experiment on the TRECVID 2011 Dataset.

References

1. Wang, H., Ding, C., Huang, H.: Multi-label linear discriminant analysis. In: ECCV. (2010)
2. Huang, D., Storer, M., De la Torre, F., Bischof, H.: Supervised local subspace learning for continuous head pose estimation. In: CVPR. (2011)
3. Huber, P.: Robust Statistics. Wiley and Sons (1981)
4. Rousseeuw, P., Leroy, A.: Robust Regression and Outlier Detection. Wiley (2003)
5. Meer, P.: Robust Techniques for computer vision, Book chapter in Emerging Topics in Computer Vision, G. Medioni and S. Kang (Eds.). Prentice Hall (2004)
6. Gillard, J.: An Historical Overview of Linear Regression with Errors in both variables. Cardiff University, School of Mathematics, TR (2006)
7. Huffel, S.V., Vandewalle, J.: The Total Least Squares Problem: Computational Aspects and Analysis. SIAM (1991)
8. Lindley, D.: Regression lines and the linear functional relationship. Suppl. J. Roy. Statist. Soc. **9** (1947) 218–244
9. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Comm. of the ACM **24** (1981) 381–395
10. Torr, P., Zisserman, A.: Mlesac: A new robust estimator with application to estimating image geometry. CVIU **78** (2000) 138–156
11. Choi, S., Kim, T., Yu, W.: Performance Evaluation of RANSAC Family. In: BMVC. (2009)
12. Adcock, R.: A problem in least squares. Analyst **5** (1878) 53–54

13. Kummel, C.: Reduction of observed equations which contain more than one observed quantity. Analyst **6** (1879) 97–105
14. Wald, A.: The fitting of straight lines if both variables are subject to error. Ann. Math. Statistics **11** (1940) 285–300
15. Gillard, J., Iles, T.: Method of moments estimation in linear regression with errors in both variables. Cardiff University, School of Mathematics, TR (2005)
16. Matei, B., Meer, P.: Estimation of nonlinear errors-in-variables models for computer vision applications. IEEE Trans. PAMI **28** (2006) 1537–1552
17. Croux, C., Dehon, C.: Robust linear discriminant analysis using s-estimators. Canadian Journal of Statistics **29** (2001)
18. Kim, S., Magnani, A., Boyd, S.: Robust FDA. In: NIPS. (2005)
19. Zhang, Y., Yeung, D.Y.: Worst-case linear discriminant analysis. In: NIPS. (2010)
20. Fidler, S., Skocaj, D., Leonardis, A.: Combining reconstructive and discriminative subspace methods for robust classification and regression by subsampling. PAMI **28** (2006) 337–350
21. Leonardis, A., Bischof, H.: Robust recognition using eigenimages. CVIU **78** (2000) 99–118
22. Jia, H., Martinez, A.: Support vector machines in face recognition with occlusions. In: CVPR. (2009)
23. De la Torre, F., Black, M.: A framework for robust subspace learning. International Journal on Computer Vision **54** (2003) 117–142
24. Candès, E., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? Journal of the ACM **58** (2011)
25. Wright, J., Ganesh, A., Rao, S., Peng, Y., Ma, Y.: Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In: NIPS. (2009)
26. Cabral, R., De la Torre, F., Costeira, J.P., Bernardino, A.: Matrix completion for multi-label image classification. In: NIPS. (2011)
27. Zhang, Z., Liang, X., Ma, Y.: Unwrapping low-rank textures on generalized cylindrical surfaces. In: ICCV. (2011)
28. Cheng, B., Liu, G., Wang, J., Huang, Z., Yan, S.: Multi-task low-rank affinity pursuit for image segmentation. In: ICCV. (2011)
29. De la Torre, F.: A least-squares framework for component analysis. IEEE Trans. PAMI **34** (2012) 1041–1055
30. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society, Series B **68** (2007) 49–67
31. Golub, G., Loan, C.V.: Regression lines and the linear functional relationship. SIAM J. Numer. Anal. **17** (1980) 883–893
32. Gross, R., Matthews, I., Cohn, J.F., Kanade, T., Baker, S.: The CMU multi-pose, illumination, and expression (multi-pie) face database. Technical report, CMU Robotics Institute.TR-07-08 (2007)
33. Snoek, C., Worring, M., Gemert, J., Geusebroek, J.M., Smeulders, A.: The challenge problem for automated detection of 101 semantic concepts in multimedia. In: ACM MM. (2006)
34. Vedaldi, A., Zisserman, A.: Efficient additive kernels via explicit feature maps. IEEE Trans. PAMI **34** (2012) 480–492
35. Li, F., Lebanon, G., Sminchisescu, C.: Chebyshev Approximations to the Histogram χ^2 Kernel. In: CVPR. (2012)