Learning Robust Representations for Computer Vision

Peng Zheng
University of Washington
Seattle, WA 98195-4322, USA
zhengp@uw.edu

Aleksandr Y. Aravkin
University of Washington
Seattle, WA 98195-4322, USA
saravkin@uw.edu

Karthikeyan Natesan Ramamurthy
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598, USA
knatesa@us.ibm.com

Jayaraman Jayaraman Thiagarajan
Lawrence Livermore National Laboratory
Livermore, CA 94550, USA
jjayaram@llnl.gov

Abstract

Unsupervised learning techniques in computer vision often require learning latent representations, such as low-dimensional linear and non-linear subspaces. Noise and outliers in the data can frustrate these approaches by obscuring the latent spaces.

Our main goal is deeper understanding and new development of robust approaches for representation learning. We provide a new interpretation for existing robust approaches and present two specific contributions: a new robust PCA approach, which can separate foreground features from dynamic background, and a novel robust spectral clustering method, that can cluster facial images with high accuracy. Both contributions show superior performance to standard methods on real-world test sets.

1. Introduction

Supervised learning, and in particular deep learning [1, 2], have been very successful in computer vision. Applications include autoencoders [3] that map between noisy and clean images [4], convolutional networks for image/video analysis [5], and generative adversarial networks that synthesize real world-like images [6].

In contrast, unsupervised learning still poses significant challenges. Broadly, unsupervised learning seeks to discover hidden structure in the data without using ground truth labels, thereby revealing features of interest.

In this paper, we consider unsupervised representation learning methods which can be used along with centroid-based clustering to summarize the data distribution using a few characteristic samples.

We are interested in spectral clustering [7] and subspace clustering [8]; the proposed ideas can also be generalized to deep embedding-based clustering strategies [9]. *Spectral clustering* methods use neighborhood graphs to learn the underlying representation [7]; this approach is used for image segmentation [10, 11] and 3D mesh segmentation [12]. *Subspace clustering* methods model the dataset as a union of low-dimensional linear subspaces and utilize sparse and low-rank methods to obtain the representation; this model is used for facial clustering and recognition [8, 13].

Learning effective latent representations hinges on accurately modeling noise and outliers. Further, in practice, the data satisfy the structural assumptions (union of subspaces, low rank, etc.) only approximately. Adopting robust optimization strategies is a natural way to combat these challenges. For example, consider principal component analysis (PCA), a prototypical representation learning method based on matrix factorization. Given low-rank data contaminated by outliers, the classical PCA method will fail to find it. Consequently, the robust PCA (rPCA) method [14], which decomposes data into low rank and sparse components, is preferred in practice, e.g. background/foreground separation [14, 15]. Similarly, when data assumed to be from a union of subspaces is contaminated by outliers, allowing for sparse outliers during optimization leads to accurate recovery of the subspaces, e.g. face classification [16].

Our goal is to develop effective robust formulations for unsupervised representation learning tasks in computer vision; we are interested in complex situations, when the data is corrupted with a combination of sparse outliers and dense noise.

Contributions. We first review the relationship between outlier models and statistically robust formulations. In particular, we show that the rPCA formulation is equivalent to solving a Huber regression problem for low-rank representation learning. Using this connection, we develop a new nonconvex penalty, dubbed the Tiber, designed to aggres-

sively penalize mid-sized residuals. In Section 2, we show that this penalty is well suited for dynamic background separation, outperforming classic rPCA methods.

Our second contribution is to use the design philosophy behind robust low-rank representation learning to develop a new formulation for robust clustering. We formulate classic spectral analysis as an optimization problem, and then modify this problem to be robust to outliers. The advantages are shown using a synthetic clustering example. We then combine robust spectral clustering with robust subspace clustering to achieve superior performance on face recognition tasks, surpassing prior work without any data pre-processing; see Section 3, Table 1.

2. New Penalties for Learning Robust Representations

Many tasks in computer vision depend on unsupervised representation learning. A well-known example is background/foreground separation, often solved by robust principal component analysis (rPCA). rPCA learns low-rank representations by decomposing a data matrix into a sum of low-rank and sparse components. The low-rank component represents the background and the sparse component represents the foreground [14].

In this section, we show that rPCA is equivalent to a robust regression problem, and solving a Huber-robust regression [17] for the background representation is completely equivalent to the full rPCA solution. We use this equivalence to design a new robust penalty (dubbed Tiber) based on statistical descriptions of the signals of interest. We illustrate the benefits of using this new non-convex penalty for separating foreground from a dynamic background, using real datasets.

2.1. Huber in rPCA

Background/foreground separation is widely used for detecting moving objects in videos from stationary cameras. A broad range of techniques have been developed to tackle this task, ranging from simple thresholding [18] to mixtures of Gaussian models[19, 20, 21]. In particular, rPCA has been widely adopted to solve this problem [22, 23].

Denote a given video stream by $Y \in \mathbb{R}^{n\times m}$, where each of m frames is reshaped to be a vector of size n. There are many variants of rPCA [24]. We use the *stable principal component pursuit* (SPCP) formulation:

$$\min_{L,S} \frac{1}{2}\|L+S-Y\|_F^2 + \kappa\|S\|_1 + \lambda\|L\|_* \qquad (1)$$

where L represents the background, and S the foreground. The regularizations used by this formulation ensure that L is chosen to be low rank, while S is designed to be sparse; using a quadratic penalty on the residual fits of the data up to some error level.

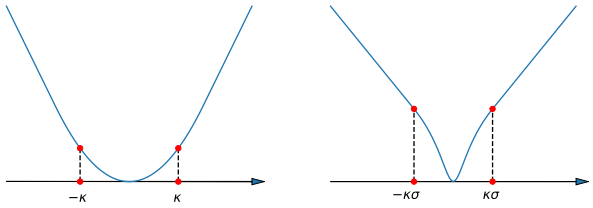

Figure 1. Robust penalties: left: Huber, right: Tiber. Both grow linearly outside an interval containing the origin. The Tiber penalizes 'mid-sized' errors within the region far more aggressively than the Huber; such a penalty must necessarily be non-convex.

We can minimize over the variables in any order. Minimizing the first two summands of (1) in S gives a closed form function

$$\min_S \frac{1}{2}\|L+S-Y\|_F^2 + \kappa\|S\|_1 = \rho(L-Y;\kappa),$$

with $\rho(r;\kappa)$ the well-known Huber penalty [17]

$$\rho(r;\kappa) = \begin{cases} \kappa|r| - \kappa^2/2, & |r| > \kappa \\ r^2/2, & |r| \le \kappa \end{cases}. \qquad (2)$$

We provide a simple statement of the following well-known result with a short self-contained proof.

Claim 1.

$$\rho(r;\kappa) = \min_s \frac{1}{2}(s-r)^2 + \kappa|s|. \qquad (3)$$

Proof. The solution to this optimization problem is the *soft threshold* function (see e.g. [25])

$$\arg\min_s \frac{1}{2}(s-r)^2 + \kappa|s| = \mathbb{S}_\kappa(r) = \begin{cases} r-\kappa, & r > \kappa \\ 0, & |r| \le \kappa \\ r+\kappa, & r < -\kappa \end{cases}.$$

Plugging $\mathbb{S}_\kappa(r)$ back into (3), we have

$$\frac{1}{2}[\mathbb{S}_\kappa(r) - r]^2 + \kappa|\mathbb{S}_\kappa(r)| = \rho(r;\kappa).$$

□

The optimization problem is separable, so the result immediately extends to the vector case. Upon minimization over S, problem (1) then reduces to

$$\min_L \rho(L-Y;\kappa) + \lambda\|L\|_*. \qquad (4)$$

To simplify the problem further, we use a factorized representation of L [26], choosing the rank to be $k \ll \min(n,m)$ to obtaining the non-convex formulation

$$\min_{U,V} \rho(U^\mathsf{T}V - Y;\kappa) \qquad (5)$$

where $U \in \mathbb{R}^{k\times n}$ and $V \in \mathbb{R}^{k\times m}$.

Comparing (5) to (1) we see two advantages:

1. The dimension of the decision variable has been reduced from $2nm$ to $k(n+m)$.
2. (5) is smooth, and does not require computing SVDs.

Once we have U and V, we can easily recover L and S:

$$L = U^{\mathsf{T}}V, \quad S = \mathbb{S}_{\kappa}(U^{\mathsf{T}}V - Y).$$

The approach is illustrated in the left panels of Figure 2. Although the residual $U^TV - Y$ (shown in row 2) is noisy and not sparse, applying $\mathbb{S}_\kappa$ we get the sparse component (row 3), just as we would by solving the original formulation (1).

From a statistical perspective, the equivalence of rPCA and Huber means that the residual $R = U^{\mathsf{T}}V - Y$, which contains both S and random noise, can be modeled by a heavy tailed error distribution.

Claim 2. *Suppose $\{r_i(x)\}_{i=1}^l$ are i.i.d. samples from a distribution with density*

$$p(r;\theta) = \frac{1}{n_c(\theta)}\exp[-\rho(r;\theta)]$$

where $n_c(\theta) = \int_{\mathbb{R}}\exp[-\rho(r;\theta)]\,dr$ is the normalization constant. Then maximum likelihood formulation for x is equivalent to the minimization problem

$$\min_x \sum_{i=1}^{l}\rho(r_i(x);\theta).$$

The claim follows immediately by taking the negative log of the maximum likelihood. Claim 2 means that solving (5) is equivalent to assuming that elements in $R = U^{\mathsf{T}}V - Y$ are i.i.d. samples from the Laplace density

$$p(r;\kappa) = \frac{1}{n_c(\kappa)}\exp[-\rho(r;\kappa)].$$

The function ρ has linear tails (See Figure 1), which means this distribution is much more likely to produce large samples compared to the Gaussian.

2.2. Weaknesses of the Huber

Although the Huber distribution can detect sparse outliers, it does not model small errors well. In many background/foreground separation problems, we must cope with a dynamic background (e.g. motion of tree leaves or water waves). These small dynamic background perturbations correspond to motion we do not care about — we are much more interested in detecting cars, people, and animals moving through the scene.

We want to move these dynamics into the low-rank background term. However, the Huber is quadratic near the origin (i.e. nearly flat), so small perturbations do not significantly affect the objective value; and solving (5) leaves these terms in the residual R. Thresholding these terms is either too aggressive (removing features we care about), or too lenient, leaving the dynamics in the foreground (see first two columns of Figure 2). A better penalty would rise steeply for small values of R, without significantly affecting tail behavior.

2.3. Tiber for rPCA

We propose a new penalty, which we call the Tiber. While the Huber is defined by partially minimizing the sum of the 1-norm with a quadratic (2), the Tiber replaces the quadratic with a nonconvex function. The resulting penalty can match the tail behavior of Huber, yet have different properties around the origin (see Figure 1). Tiber is better suited for background/foreground separation problems with dynamic background. We define the penalty as follows:

$$\rho_{\mathrm{T}}(r;[\kappa,\sigma]) = \begin{cases} \frac{2\kappa}{\sigma(\kappa^2+1)}(|r| - \kappa\sigma) + \log(1+\kappa^2), & |r| > \kappa\sigma \\ \log(1 + r^2/\sigma^2), & |r| \le \kappa\sigma \end{cases} \quad (6)$$

The Tiber is parametrized by thresholding parameter κ and scale parameter σ. Just as the Huber, it can be expressed as the value function of a minimization problem. We replace the quadratic penalty in Claim 1 by the smooth nonconvex penalty $\log(1+(\cdot)^2)$. For simplicity, we use $\sigma = 1$ in the result below.

Claim 3.

$$\rho_{\mathrm{T}}(r;[\kappa,1]) = \min_s \log(1+(s-r)^2) + \frac{2\kappa}{1+\kappa^2}|s|. \quad (7)$$

Proof. Denote the objective function in (7) by $f(s)$. It is easy to check that f is quasi-convex in s when $\kappa \ge 0$. We look to local optimality conditions to understand the structure of the minimizers.

- Suppose $s^* > 0$. Then $0 = f'(s^*)$ means

  $$0 = \frac{2(s^*-r)}{1+(s^*-r)^2} + \frac{2\kappa}{1+\kappa^2} \iff s^* = r - \kappa;$$

  this requires $r > \kappa$.

- Suppose $s^* < 0$. Then $0 = f'(s^*)$ means

  $$0 = \frac{2(s^*-r)}{1+(s^*-r)^2} + \frac{-2\kappa}{1+\kappa^2} \iff s^* = r + \kappa;$$

  this requires $r < -\kappa$.

- otherwise $s^* = 0$.

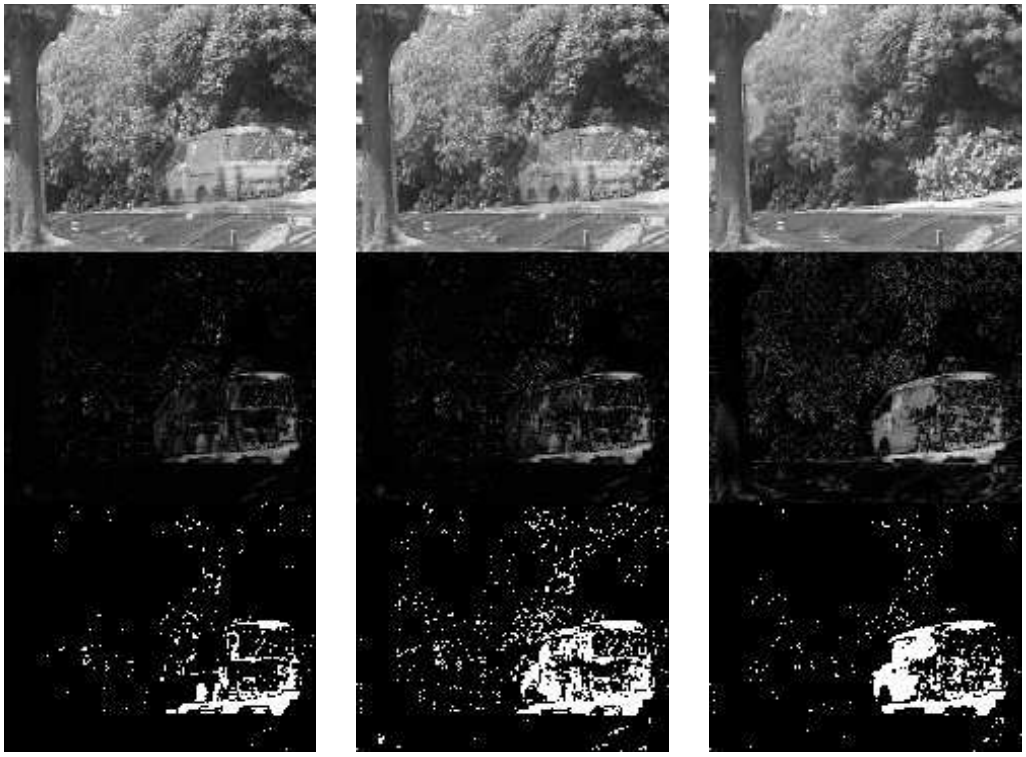

Figure 2. Left: Huber with $\kappa = 0.15$, middle: Huber with $\kappa = 0.1$, right: Tiber with $\kappa = 10, \sigma = 0.03$. Row 1: low rank component L, row 2: residual $|R| = |U^\mathsf{T}V - Y|$, row 3: binary plot for S. The Tiber recovers the van while avoiding the dynamic background.

Therefore $s^* = \mathbb{S}_\kappa(r)$. Plugging this into (7), we have

$$\rho_\mathrm{T}(r;[\kappa,1]) = \log(1 + (\mathbb{S}_\kappa(r) - r)^2) + \frac{2\kappa}{1+\kappa^2}|\mathbb{S}_\kappa(r)|.$$

□

In Figure 1, we see that Tiber rises steeply near the origin. This behavior discourages dynamic terms (leaves, waves) in R, forcing them to be fit by $U^\mathsf{T}V$. The new Tiber-robust rPCA problem is given by:

$$\min_{U,V} \rho_\mathrm{T}(U^\mathsf{T}V - Y;[\kappa,\sigma]) \tag{8}$$

which also has all of the advantages of (5). Moreover, because of the characterization from Claim 3, once we solve (8), we immediately recover L and S:

$$L = U^\mathsf{T}V, \quad S = \mathbb{S}_{\kappa\sigma}(U^\mathsf{T}V - Y).$$

2.4. Experiment: Foreground Separation

We use a publicly available data set[1] with a dynamic background (moving trees). We sample 102 data frames from this data set, convert them to grey scale, and reshape them as column vectors of matrix $Y \in \mathbb{R}^{20480\times102}$. We compare formulations (5) and (8). Proximal alternating linearized minimization algorithm (PALM) [27] was used to solve all of the optimization problems.

Rank of U and V was set to be $k = 10$ for all experiments. We manually tuned parameters to achieve the best possible recovery in each formulation. For Huber, we selected two nearby κ values, $\kappa = 0.15$ and $\kappa = 0.1$; for Tiber, we selected $\kappa = 10$ and $\sigma = 0.03$, resulting in the threshold parameter $\kappa\sigma = 0.3$.

The results are shown in Figure 2. The task is identifying the van while avoiding interference from moving leaves. The Huber is unable to separate the van from the leaves for any threshold values κ. When we choose $\kappa = 0.15$ (left panel in Figure 2), we cut out too much information, giving an incomplete van in S. If we make a less conservative choice $\kappa = 0.1$ (middle panel in Figure 2), we leave too much dynamic noise in S, which obscures the van.

The Tiber Penalty obtains a cleaner picture of the moving vehicle (right panel in Figure 2). As expected, it forces more of the dynamic background to be fit by $U^\mathsf{T}V$, leaving

[1]Downloaded from http://vis-www.cs.umass.edu/~narayana/castanza/I2Rdataset/

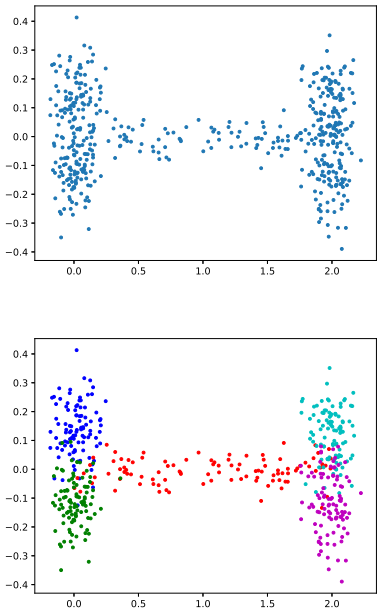

Figure 3. Synthetic Data Clustering: Up: data without labels, Down: data with true colors.

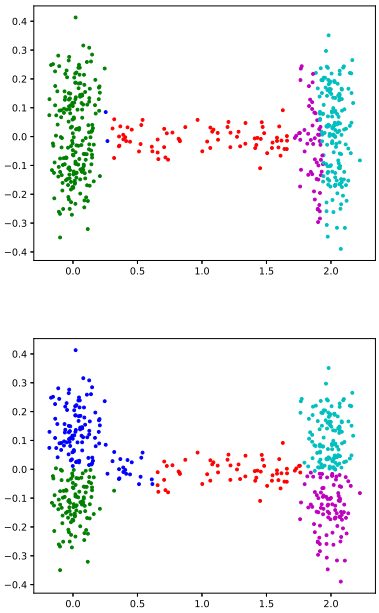

Figure 4. Synthetic Data Clustering: Up: result from eigenvalue decomposition, Down: result from (10).

a fairly complete van in S without too much contamination.

3. Robust Representation Learning for Clustering

Centroid-based clustering, e.g. k-Means, is a standard tool to partition and summarize datasets. Given the high dimensionality and complexity of data in computer vision applications, it is necessary to learn latent representations, such as the underlying metric, prior to clustering. Clustering is then performed in the latent space.

We develop an approach for robust spectral clustering. We illustrate the advantages using a synthetic dataset, and then combine the approach with robust subspace clustering to achieve perfect performance on face recognition tasks.

3.1. Spectral Clustering

Spectral clustering [7] is formulated as follows. Given m datapoints $y_i \in \mathbb{R}^n$, we arrange them in a matrix $Y \in \mathbb{R}^{n \times m}$. To partition the data into k groups, spectral clustering uses the following steps:

1. Given a dataset of m samples, we construct the similarity matrix $L \in \mathbb{R}^{m \times m}$ of the data points.
2. Extract the eigenvectors $X \in \mathbb{R}^{m \times k}$ of L corresponding to the k largest eigenvalues.
3. Project each row of X onto the unit ball, and apply distance-based clustering.

Finding a meaningful similarity matrix L is crucial to the success of spectral clustering. Ideally, L will be a block diagonal matrix with k blocks. This rarely happens for real applications; even when underlying structure in L is present, it can be obscured by noise and a small number of points that don't follow the general pattern.

To find a factorization of noisy L, we need a robust method for eigenvalue decomposition. We first formulate eigenvalue decomposition as an optimization problem.

Claim 4. *Assume L is a symmetric matrix with eigenvalues less than or equal to 1. Then the solution to the problem*

$$\begin{aligned} &\min_X \ \frac{1}{2}\|XX^\mathsf{T} - L\|_F^2 \\ &\textit{s.t. } X^\mathsf{T}X = I_k \end{aligned} \tag{9}$$

is $X = [v_1, \ldots, v_k]$ with v_i the eigenvector corresponding to the i^{th} largest eigenvalue of L, and I_k the k by k identity matrix.

Proof. Since L is a symmetric matrix, it has a eigenvalue decomposition,

$$L = Y \Lambda Y^\mathsf{T}$$

where $Y \in \mathbb{R}^{m \times m}$ is orthogonal and Λ is diagonal, with $1 \geq \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$. Similarly, we have

$$XX^\mathsf{T} = \tilde{X} D \tilde{X}^\mathsf{T}$$

where $\tilde{X} \in \mathbb{R}^{m\times m}$ is a orthogonal matrix whose first k columns agree with those of X, D is a diagonal matrix with first k elements on the diagonal are 1 and the rest are 0. From the Cauchy-Schwarz inequality, we have

$$\text{trace}\,(XX^\mathsf{T}\cdot L) = \langle XX^\mathsf{T}, L\rangle \le \|XX^\mathsf{T}\|_F \cdot \|L\|_F$$

where equality hold when XX^T and L share the same singular vectors, i.e., X equals to the first k columns of Y. Therefore

$$\begin{aligned}\frac{1}{2}\|XX^\mathsf{T} - L\|_F^2 &= \frac{1}{2}\|XX^\mathsf{T}\|_F^2 - \langle XX^\mathsf{T}, L\rangle + \frac{1}{2}\|L\|_F^2 \\ &\ge \frac{1}{2}\|D\|_F^2 - \|D\|_F\|\Lambda\|_F + \frac{1}{2}\|\Lambda\|_F^2\end{aligned}$$

with equality hold when columns of X are eigenvectors corresponding to the largest k eigenvalues. □

We robustify (9) by replacing the Frobenius norm in the optimization formulation by the Huber function (or another robust penalty):

$$\begin{aligned}&\min_X\ \rho(XX^\mathsf{T} - L;\kappa) \\ &\text{s.t. } X^\mathsf{T}X = I_k\end{aligned}. \tag{10}$$

This approach can be very effective. Consider the following clustering experiment with $n = 2$, $m = 500$, and $k = 5$. We generate five clusters (sampling from four 2-D Gaussians, one rectangular uniform distribution) with 100 points per group. To make the problem challenging, we move the clusters close together so much that trying to tell them apart with the naked eye is hard (Figure 3, top). True clusters appear in Figure 3, bottom.

Classic spectral clustering, which uses eigenvalue decomposition in step 2, fails to detect the true relationships (Figure 4, top). Robust spectral clustering using the Huber penalty (10) does a much better job (Figure 4, bottom).

3.2. Subspace Clustering

Subspace clustering looks for low dimensional representation of high dimensional data, by grouping the points along low-dimensional subspaces. Given a data matrix $Y \in \mathbb{R}^{n\times m}$ as in Section 3.1, the optimization for subspace clustering is given by [8]:

$$\min_C\ \frac{1}{2}\|Y - YC\|_F^2 + \lambda\|C\|_1\ \text{ s.t. }\ \text{diag}\,(C) = 0. \tag{11}$$

This formulation looks for a sparse representation of the dataset by its members: $s_i = Sc_i$. To avoid the trivial solution, we require the diagonal of C to be identically 0. After obtaining C, it is post-processed and a similarity matrix is constructed as $W = |C| + |C^\mathsf{T}|$. W will be ideally close to block-diagonal, where each block represents a subspace, and spectral clustering is performed it to identify cluster memberships.

Outliers in the dataset can break the performance of (11). To make the approach robust, [8] uses the formulation

$$\begin{aligned}&\min_C\ \frac{1}{2}\|Y - YC - S\|_F^2 + \lambda\|C\|_1 + \kappa\|S\|_1 \\ &\text{s.t. diag}\,(C) = 0.\end{aligned} \tag{12}$$

Using Claim 1, we rewrite (12) using Huber:

$$\min_C\ \rho(Y - YC;\kappa) + \lambda\|C\|_1\ \text{ s.t. }\ \text{diag}\,(C) = 0. \tag{13}$$

Formulation (13) has the same advantages with respect to (12) as (5) has with respect to (1).

3.3. Face Clustering

Given multiple face images taken at different conditions, the goal of face clustering [8] is to identify images that belong to the same person.

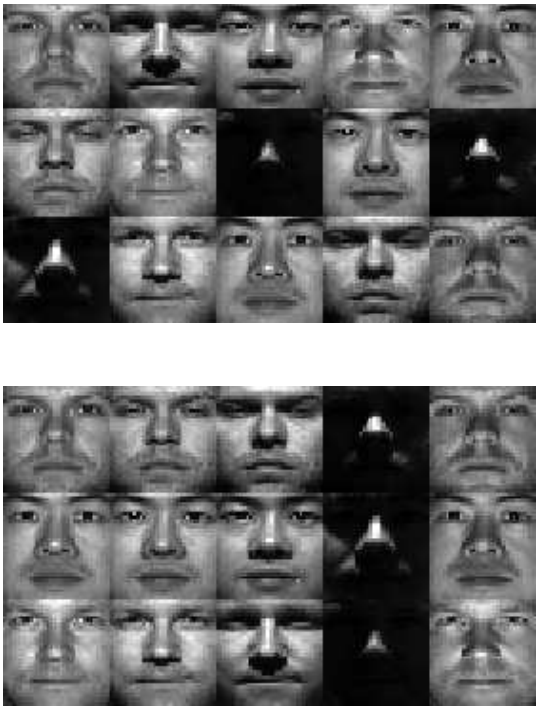

Figure 5. Faces data: top: randomly chosen face images, bottom: faces after clustering; each row belongs to a cluster.

We use images from the publicly available Extended Yale B dataset [28] [2]. Each image has 32×32 pixels, and

[2]Downloaded from http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html

there are 2414 images in the dataset. These images belong to 38 people, with approximately 64 pictures per person.

Under the Lambertian assumption, pictures obtained from one person under different illuminations should lie close to a 9 dimensional subspace [29]. In practice, these spaces are hard to detect because of noise in the images, and a robust approach is required.

Robust subspace clustering for face images:

1. Obtain sparse representation C using (13).
2. Construct similarity matrix W from C.
   - Normalize columns of C to have maximum absolute value no larger than 1.
   - Form $W = |C| + |C^\mathsf{T}|$
   - Normalize W: $W \leftarrow D^{-1/2} W D^{-1/2}$, where D is a diagonal matrix with $D_{ii} = \sum_j W_{ij}$.
3. Apply spectral clustering using W.
   - Apply robust symmetric factorization (10) to W, to obtain the latent representation X.
   - Project each row of X onto the unit 2-norm ball.
   - Apply K-means algorithm to the new rows of X.

The results are shown in Table 1. We implement the approach for different numbers of subjects $k = 2, 3, 5, 8$. We show the parameters κ and λ in (13) used to achieve the high accuracies given in Table 1[3].

clusters	κ in (13)	λ in (13)	error	error in [8]
$k = 2$	0.5	1	0.00%	1.86%
$k = 3$	0.1	0.7	0.00%	3.10%
$k = 5$	0.05	0.7	0.00%	4.31%
$k = 8$	0.03	0.5	2.73%	5.85%

Table 1. Results for robust subspace clustering with face images.

To get better intuition of the method, we plot the similarity matrix corresponding to $k = 3$ in Figure 6. We can clearly see three blocks along the diagonal that correspond to the three face clusters. The resulting projected X obtained from the eigenvalue decomposition of similarity matrix W are shown in Figure 7. The three clusters are clearly well separated. The final algorithm has perfect accuracy in this example.

4. Discussion

Robust approaches are essential for unsupervised learning, and can be designed using optimization formulations. For example, in both rPCA and robust spectral learning, SVD and eigenvalue decomposition are first characterized using optimization, then reformulated with robust losses.

[3]In [8], the images used are of size 48×42. The numbers shown are therefore indicative.

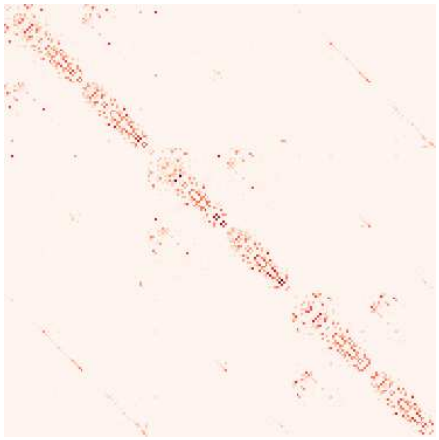

Figure 6. Similarity matrix for face images clustering with $k = 3$; the matrix is nearly block diagonal with 3 blocks.

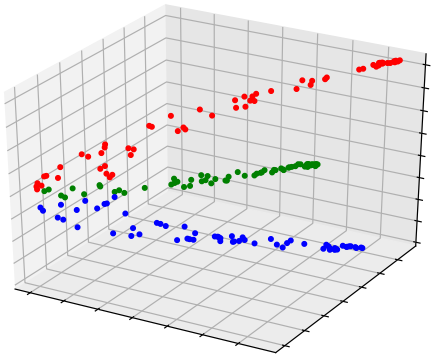

Figure 7. Projections of the rows of X onto the eigenspace of the similarity matrix for $k = 3$. Each color represent the face images of a single person.

Several tasks in this approach are difficult. First, there is a need to tune parameters in the optimization formulations. For example, the Tiber depends on two parameters, κ and σ. Automatic ways to tune these parameters can make robust unsupervised learning a lot more portable. Second, the optimization problems we have to solve are large-scale; time required for robust subspace clustering for images scales non-linearly with both the number and size of images. Designing non-smooth stochastic algorithms that take the structure of these problems into account is essential.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. 1

[2] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015. 1

[3] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010. 1

[4] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 341–349. 1

[5] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732. 1

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680. 1

[7] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856. 1, 5

[8] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013. 1, 6, 7

[9] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International Conference on Machine Learning*, 2016, pp. 478–487. 1

[10] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000. 1

[11] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in neural information processing systems*, 2005, pp. 1601–1608. 1

[12] R. Liu and H. Zhang, "Segmentation of 3d meshes through spectral clustering," in *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on*. IEEE, 2004, pp. 298–305. 1

[13] G. Shakhnarovich and B. Moghaddam, "Face recognition in subspaces," in *Handbook of Face Recognition*. Springer, 2011, pp. 19–49. 1

[14] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011. 1, 2

[15] A. Sobral, T. Bouwmans, and E.-h. Zahzah, "Lrslibrary: Low-rank and sparse tools for background modeling and subtraction in videos," *Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing. CRC Press, Taylor and Francis Group*, 2016. 1

[16] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 2, pp. 210–227, 2009. 1

[17] P. J. Huber, "Robust statistics," in *International Encyclopedia of Statistical Science*. Springer, 2011, pp. 1248–1251. 2

[18] T. Veit, F. Cao, and P. Bouthemy, "A maximality principle applied to a contrario motion detection," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 1. IEEE, 2005, pp. I–1061. 2

[19] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2. IEEE, 1999, pp. 246–252. 2

[20] R. H. Evangelio, M. Pätzold, and T. Sikora, "Splitting gaussians in mixture models," in *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*. IEEE, 2012, pp. 300–305. 2

[21] T. S. Haines and T. Xiang, "Background subtraction with dirichlet processes," in *European Conference on Computer Vision*. Springer, 2012, pp. 99–113. 2

[22] C. Guyon, T. Bouwmans, and E.-h. Zahzah, "Robust principal component analysis for background subtraction: Systematic evaluation and comparative analysis," in *Principal component analysis*. InTech, 2012. 2

[23] R. Otazo, E. Candès, and D. K. Sodickson, "Low-rank plus sparse matrix decomposition for accelerated dynamic mri with separation of background and dynamic components," *Magnetic Resonance in Medicine*, vol. 73, no. 3, pp. 1125–1136, 2015. 2

[24] A. Aravkin and S. Becker, "Dual smoothing and value function techniques for variational matrix decomposition," in *Handbook of Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*, T. Bouwmans, N. S. Aybat, and E.-h. Zahzah, Eds. CRC Press, 2016, ch. 3. 2

[25] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*. Springer, 2011, pp. 185–212. 2

[26] S. Burer and R. D. Monteiro, "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization," *Mathematical Programming*, vol. 95, no. 2, pp. 329–357, 2003. 2

[27] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1-2, pp. 459–494, 2014. 4

[28] K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 5, pp. 684–698, 2005. 6

[29] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE transactions on pattern analysis and machine intelligence*, vol. 25, no. 2, pp. 218–233, 2003. 7