SPARSE SUBSPACE CLUSTERING USING SQUARE-ROOT PENALTY

Linghang Meng, Xinyue Shen, and Yuantao Gu

Tsinghua National Laboratory for Information Science and Technology
Department of Electronic Engineering, Tsinghua University, Beijing 100084, CHINA

ABSTRACT

We study the sparse subspace clustering problem in presence of both sparse outliers and Gaussian additive noise based on data sparse self-representation. We propose a convex optimization problem which does not only induce sparsity on the representation coefficients and the outliers, but also adopts a square-root penalty to improve the robustness against Gaussian noise. An algorithm based on alternating direction method of multipliers (ADMM) is then devised as a solver for the proposed problem. As a real application, the proposed model and algorithm are applied in motion segmentation. The performances are demonstrated and analyzed by synthetic data, and more importantly, the effectiveness is verified by some real data. Compared with the reference method, numerical results show that the new method achieves higher cluster accuracy and that the choice of the parameter can be less sensitive to the noise level. [1]

Index Terms— sparse subspace clustering, sparse representation, square-root penalty, ADMM, motion segmentation

1. INTRODUCTION

High-dimensional data are ubiquitous in many areas of machine learning and signal processing, such as computer vision [1, 2], pattern recognition [3], and bioinformations [4]. The high-dimensionality does not only increase the computational complexity of algorithms, but also adversely affects their performances due to the noise effect and the insufficient number of samples compared to the ambient space dimension [5]. A help comes with the fact that in many cases high-dimensional data approximately lie in low-dimensional structures [6, 7], and among them subspace and union of subspaces are typical and useful ones [8, 9]. In the light of such fact, as an unsupervised learning model, subspace clustering has demonstrated its power in motion segmentation [10], face clustering [11], and handwritten digit detection [12].

The formulation of subspace clustering goes as follows. Let $\{\mathcal{S}_l\}_{l=1}^n$ be a collection of n linear subspaces in $\mathbb{R}^D$, each of dimension d_l. Noise-free data points $\{\mathbf{x}_i\}_{i=1}^N$ lie in the union $\cup_{l=1}^n \mathcal{S}_l$. In the presence of noise and sparse outlying entries, let

$$\mathbf{y}_i = \mathbf{x}_i + \mathbf{e}_i + \mathbf{z}_i$$

be the ith observation obtained by corrupting an error-free point $\mathbf{x}_i$. The vector of sparse outlying entries $\mathbf{e}_i \in \mathbb{R}^D$ has only $k \ll D$ nonzero elements, and $\mathbf{z}_i$ obeys a Gaussian distribution $\mathcal{N}\left(0, \sigma_z^2 \mathbf{I}\right)$ with unknown variance σ_z. Denote $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$. Subspace clustering refers to the problem of finding the number of subspaces n, the dimensions d_l for $l = 1, \ldots, n$, the bases for these subspaces, and the segmentation of the data, given only the corrupted observation $\mathbf{Y}$.

Sparse subspace clustering (SSC) uses the idea that $\mathbf{X}$ is a self-expressive dictionary in which each point can be written as a linear combination of a few other points [9], and has led to a family of subspace clustering methods [13, 14]. In general, its process includes two steps, sparse self-representation and spectral clustering. In the first step, considering noise and sparse outlying entries, a sparse representation can be estimated by solving the following convex problem [14]

$$\begin{array}{ll} \text{minimize} & \|\mathbf{C}\|_1 + \lambda_e \|\mathbf{E}\|_1 + \lambda_z \|\mathbf{Z}\|_F^2 \\ \text{subject to} & \mathbf{Y} = \mathbf{Y}\mathbf{C} + \mathbf{E} + \mathbf{Z}, \quad \operatorname{diag}(\mathbf{C}) = \mathbf{0}, \end{array} \tag{1}$$

where $\mathbf{C}$, $\mathbf{E}$, and $\mathbf{Z}$ are variables, and they denote the matrix of self-representation coefficients, the impulse noise, and the Gaussian noise, respectively. In [9] for the same purpose a problem is proposed with $\lambda_e = 1$, and $\|\mathbf{Z}\|_F$ is adopted instead of $\|\mathbf{Z}\|_F^2$. The solution of $\mathbf{C}$ gives a sparse representation of each data point by other points, and its nonzero entries correspond to points in the same subspace. In the second step, an N-vertex undirected graph with edge weight $\mathbf{W}$ is constructed as follows

$$\mathbf{W} = |\mathbf{C}|^{\mathrm{T}} + |\mathbf{C}|. \tag{2}$$

Then spectral clustering [15] is adopted to partition the data points into clusters according to $\mathbf{W}$.

However, for (1) the noise level should be a prior, in that the choice of the regularization parameter λ_z, which is crucial to balance the sparsity-inducing terms and noise penalty, is determined by the noise power. Recently, square-root Lasso has been proposed to replace the square of the ℓ_2 norm in Lasso with the ℓ_2 norm itself. The advantage is that the choice of the regularization parameter is independent of the additive noise power [16], and under some conditions the optimal regularization parameter can even be analytically determined [17]. Enlightened by square-root Lasso, we propose to use a square-root term to penalize the Gaussian error, in the hope that the choice of λ_z could be less sensitive to the noise level. Note that square-root penalty has previously been applied in SSC [9], but in this work we will focus more on the potential advantage mentioned above brought by the square-root term.

In this paper, we study the sparse subspace clustering problem in presence of both sparse outliers and Gaussian additive noise based on data sparse self-representation. In section 2, an optimization problem is proposed to induce the sparsity on the representation coefficients and the outliers by ℓ_1 terms and improve the robustness against Gaussian noise by a square root term. An algorithm based on alternating direction method of multipliers (ADMM) [18] is devised as a solver. In section 3, new data points are classified into

[1] This work was partially supported by National Natural Science Foundation of China (NSFC 61371137, 61531166005, 61571263, 51459003) and Tsinghua University Initiative Scientific Research Program (Grant 2014Z01005). The corresponding author of this work is Y. Gu (E-mail: gyt@tsinghua.edu.cn).

one of the subspaces learned by the proposed model and algorithm. The effectiveness is verified by both synthetic data and various real datasets in section 4. The results show that our method outperforms the reference methods in terms of clustering accuracy and parameter sensitivity to the noise level. We conclude this work in section 5.

2. SPARSE SUBSPACE CLUSTERING WITH SQUARE-ROOT PENALTY

We propose to solve the robust sparse subspace clustering problem by the following optimization problem

$$\begin{array}{ll}\text{minimize} & \|\mathbf{C}\|_1+\lambda_e\|\mathbf{E}\|_1+\lambda_z\|\mathbf{Z}\|_F \\ \text{subject to} & \mathbf{Y}=\mathbf{YC}+\mathbf{E}+\mathbf{Z},\quad \operatorname{diag}(\mathbf{C})=\mathbf{0},\end{array} \tag{3}$$

where $\mathbf{C}$, $\mathbf{E}$, and $\mathbf{Z}$ are variables. The ℓ_1 norm terms in the objective function are to induce the sparsity of the self-representation matrix and the outliers, and the F-norm of $\mathbf{Z}$, rather than its square, is used in the hope that by eliminating the square the choice of the parameter λ_z could be independent of the strength of the additive Gaussian noise.

The optimization problem (3) is convex, and we use a method based on ADMM to solve it. To begin with, we introduce an auxiliary variable and obtain the following optimization problem equivalent to (3)

$$\begin{array}{ll}\text{minimize} & \|\mathbf{C}\|_1+\lambda_e\|\mathbf{E}\|_1+\lambda_z\|\mathbf{Z}\|_F \\ \text{subject to} & \mathbf{Y}=\mathbf{YA}+\mathbf{E}+\mathbf{Z},\ \operatorname{diag}(\mathbf{C})=\mathbf{0},\ \mathbf{A}=\mathbf{C},\end{array} \tag{4}$$

where $\mathbf{A}$ is an introduced variable. An augmented Lagrangian function of (4) is

$$\begin{aligned}&L(\mathbf{A},\mathbf{C},\mathbf{E},\mathbf{Z},\mathbf{\Lambda}_1,\mathbf{\Lambda}_2,\mathbf{\Lambda}_3)\\ =&\|\mathbf{C}\|_1+\lambda_e\|\mathbf{E}\|_1+\lambda_z\|\mathbf{Z}\|_F+\langle\mathbf{\Lambda}_1,\mathbf{Y}-\mathbf{YA}-\mathbf{E}-\mathbf{Z}\rangle\\ &+\frac{\rho_1}{2}\|\mathbf{Y}-\mathbf{YA}-\mathbf{E}-\mathbf{Z}\|_F^2+\langle\mathbf{\Lambda}_2,\mathbf{A}-\mathbf{C}\rangle\\ &+\frac{\rho_2}{2}\|\mathbf{A}-\mathbf{C}\|_F^2+\langle\mathbf{\Lambda}_3,\operatorname{diag}(\mathbf{C})\rangle+\frac{\rho_3}{2}\|\operatorname{diag}(\mathbf{C})\|_F^2.\end{aligned}$$

In the iterations, we first update $\mathbf{A}$ by fixing the other variables and minimize $L(\cdot)$, and the result has the following closed form

$$\begin{aligned}\mathbf{A}^{t+1}=(\rho_1\mathbf{Y}^{\mathrm{T}}\mathbf{Y}+\rho_2\mathbf{I})^{-1}\Big(&\rho_2\mathbf{C}^t+\mathbf{Y}^{\mathrm{T}}\mathbf{\Lambda}_1^t\\ &-\rho_1\mathbf{Y}^{\mathrm{T}}(\mathbf{E}^t+\mathbf{Z}^t-\mathbf{Y})-\mathbf{\Lambda}_2^t\Big).\end{aligned} \tag{5}$$

Then $\mathbf{E}$ is updated as the following

$$\mathbf{E}^{t+1}=\operatorname{prox}_{\frac{\lambda_e}{\rho_1}\|\cdot\|_1}\left(\mathbf{Y}-\mathbf{YA}^{t+1}-\mathbf{Z}^t+\frac{1}{\rho_1}\mathbf{\Lambda}_1^t\right). \tag{6}$$

At the same time, $\mathbf{C}$ can be updated in parallel with $\mathbf{E}$ as the following

$$\begin{aligned}\operatorname{diag}(\mathbf{C}^{t+1})=\operatorname{prox}_{\frac{1}{\rho_2+\rho_3}\|\cdot\|_1}\Big(&\frac{\rho_2}{\rho_2+\rho_3}\operatorname{diag}(\mathbf{A}^{t+1})-\\ &\frac{1}{\rho_2+\rho_3}\operatorname{diag}(\mathbf{\Lambda}_3^t-\mathbf{\Lambda}_2^t)\Big)\end{aligned} \tag{7}$$

$$\begin{aligned}\operatorname{nondiag}(\mathbf{C}^{t+1})=\operatorname{prox}_{\frac{1}{\rho_2}\|\cdot\|_1}\Big(&\frac{1}{\rho_2}\operatorname{nondiag}(\mathbf{\Lambda}_2^t)+\\ &\operatorname{nondiag}(\mathbf{A}^{t+1})\Big).\end{aligned} \tag{8}$$

The update of $\mathbf{Z}$ is as the following

$$\mathbf{Z}^{t+1}=\operatorname{prox}_{\frac{\lambda_z}{\rho_1}\|\cdot\|_F}\left(\mathbf{Y}-\mathbf{YA}^{t+1}-\mathbf{E}^{t+1}+\frac{1}{\rho_1}\mathbf{\Lambda}_1^t\right). \tag{9}$$

Then the dual variables ascend as the following

$$\mathbf{\Lambda}_1^{t+1}=\mathbf{\Lambda}_1^t+\rho_1(\mathbf{Y}-\mathbf{YA}^{t+1}-\mathbf{E}^{t+1}-\mathbf{Z}^{t+1}), \tag{10}$$

$$\mathbf{\Lambda}_2^{t+1}=\mathbf{\Lambda}_2^t+\rho_2(\mathbf{A}^{t+1}-\mathbf{C}^{t+1}), \tag{11}$$

$$\operatorname{diag}(\mathbf{\Lambda}_3^{t+1})=\operatorname{diag}(\mathbf{\Lambda}_3^t)+\rho_3\operatorname{diag}(\mathbf{C}^{t+1}). \tag{12}$$

The operator $\operatorname{diag}(\cdot)$ and $\operatorname{nondiag}(\cdot)$ refer to the diagonal and non-diagonal elements, respectively. Note that all proximal operators here are element-wise. The proximal operator of the vector ℓ_1 norm is

$$\mathbf{P}=\operatorname{prox}_{\tau\|\cdot\|_1}(\mathbf{A})=\underset{\tilde{\mathbf{P}}}{\operatorname{argmin}}\frac{1}{2}\|\tilde{\mathbf{P}}-\mathbf{A}\|_F^2+\tau\|\tilde{\mathbf{P}}\|_1,$$

where $p_{ij}=\operatorname{sign}(a_{ij})\max(|a_{ij}|-\tau,0)$, and the matrix in the ℓ_1 norm is understood as a vector. The proximal operator of the F-norm of a matrix is

$$\mathbf{P}=\operatorname{prox}_{\tau\|\cdot\|_F}(\mathbf{A})=\underset{\tilde{\mathbf{P}}}{\operatorname{argmin}}\frac{1}{2}\|\tilde{\mathbf{P}}-\mathbf{A}\|_F^2+\tau\|\tilde{\mathbf{P}}\|_F,$$

where

$$\mathbf{P}=\beta\mathbf{A},\beta=\max\left(0,\frac{\|\mathbf{A}\|_F^2-\tau\|\mathbf{A}\|_F}{\|\mathbf{A}\|_F^2}\right).$$

It should be mentioned that in [9] an optimization problem similar to (3) is proposed and solved by a generic convex program solver. The differences are that they fix $\lambda_e=1$, while we give one more dimension of freedom to the method by introducing λ_e, and that we use an ADMM algorithm to solve it. In [14] an ADMM method is proposed to solve problem (1), while here we use a square-root term in (3).

3. CLASSIFICATION VIA SSC

In this section, we use the result obtained from the above SSC method to classify new data points. Our method includes two phases. The first one is to learn a basis of each subspace according to a solution to (3). The second phase is to determine which class an input sample belongs to. The detailed procedure is summarized in Algorithm 1.

In the first phase, all samples in dataset $\mathbf{Y}$ are labeled by their subspaces according to a solution to problem (3) obtained by the proposed method in the previous section. After subspace clustering, we know the subspaces for each category. According to the fact that the dimensions of the category subspaces are usually much lower than the ambient space, we may apply PCA to each of the category subspaces and get their principal components. Assume that $\mathbf{U}_1,\ldots,\mathbf{U}_n\in\mathbb{R}^{D\times k}$ are the principal components for training sets $\mathbf{Y}_1,\ldots,\mathbf{Y}_n$, respectively. To find the label of a new data point $\mathbf{x}$, we project it along each category's first k principal components, and the label is determined as

$$\text{label}(\mathbf{x})=\underset{i}{\operatorname{argmax}}\|\mathbf{U}_i^{\mathrm{T}}\mathbf{x}\|_2. \tag{13}$$

In this way, we only need to project a new data point n times to determine its label.

Algorithm 1 The Procedure of SR-SSC

Input: Dataset matrix $\mathbf{Y} \in \mathbb{R}^{D\times N}$, query point $\mathbf{x} \in \mathbb{R}^D$, parameters k, λ_e, λ_z and parameters ρ_1, ρ_2, ρ_3 in ADMM.

Output: Labels of points $\mathbf{y}_i (i = 1, \ldots, N)$ in $\mathbf{Y}$ and $\mathbf{x}$.

Phase 1. Sparse Subspace Clustering

Step 1. Estimate the sparse representation for all $\mathbf{y}_i$.

Initialize: $t = 0$, $\mathbf{A}^0, \mathbf{E}^0, \mathbf{C}^0, \mathbf{Z}^0, \mathbf{\Lambda}_1^0, \mathbf{\Lambda}_2^0, \mathbf{\Lambda}_3^0$;

while stop criterion not satisfied **do**

Update $\mathbf{A}^{t+1}$ according to (5);

Update $\mathbf{E}^{t+1}$ according to (6);

Update $\mathbf{C}^{t+1}$ according to (7), (8);

Update $\mathbf{Z}^{t+1}$ according to (9);

Update $\mathbf{\Lambda}_1^{t+1}$, $\mathbf{\Lambda}_2^{t+1}$, $\mathbf{\Lambda}_3^{t+1}$ according to (10), (11), (12);

$t = t + 1$;

end while

Step 2. Obtain subspaces.

Construct graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ by (2);

Apply spectral clustering to obtain the partition of $\mathbf{Y}$;

Perform PCA and obtain $\mathbf{U}_1, \cdots, \mathbf{U}_n \in \mathbb{R}^{D\times k}$.

Phase 2. Subspace Classification

Step 3. Determine the label for $\mathbf{x}$ by (13).

4. NUMERICAL EXPERIMENTS

The proposed method are evaluated by synthetic data and real-world datasets, and the performances are shown and studied in all these cases [2].

4.1. Synthetic data

We first generate the a set of unit orthogonal vectors $\mathcal{A}$ that span the shared dimensions among different subspaces, then the sets of unit orthogonal vectors $\mathcal{B}_i$ for the categories. Sets $\mathcal{A}$ and $\mathcal{B}_i$ together form a basis for a subspace. When generating samples in the subspaces, we generate unit coefficient vectors randomly, and multiply these coefficients by the basis. If needed, we also add Gaussian noise to each synthetic sample and then normalize it, and generate outliers drawn from the unit sphere to the synthetic dataset.

In the first experiment, the proposed method is compared with the SSC algorithm [14] under different noise levels. To have a fair comparison between our square root model and the method in [14], which does not consider outliers, in our model we set $\lambda_e = 100$, which is very large and leads solutions of $\mathbf{E}$ to 0, so the effect of the sparse outlier penalty is eliminated. The dimensions of ambient space and each subspace are 30 and 5, respectively, and there is no intersection between subspaces, i.e., $\mathcal{A} = \emptyset$. The noise to signal ratio varies from 0 to 1.5. When there are 3, 4, or 5 subspaces, the data are generated and tested for 100 times, where the parameter for both algorithms are elaborately selected as the optimal. According to the average clustering error demonstrated in Fig. 1, the proposed method outperforms the SSC algorithm, especially when the noise is strong. To be specific, when the noise to signal ratio is 0.3, the error rate of SSC is higher than 10%, while our method still achieves error rate lower than 5%.

[2]The MATLAB codes for the proposed methods and all experiments are available at `http://gu.ee.tsinghua.edu.cn/codes/SSC_SRP.zip`.

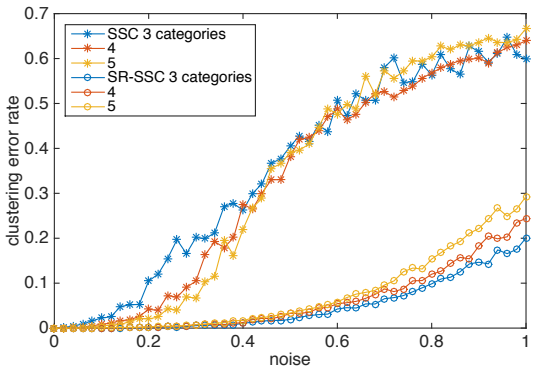


Fig. 1. Average clustering error in the first experiment for synthetic data.

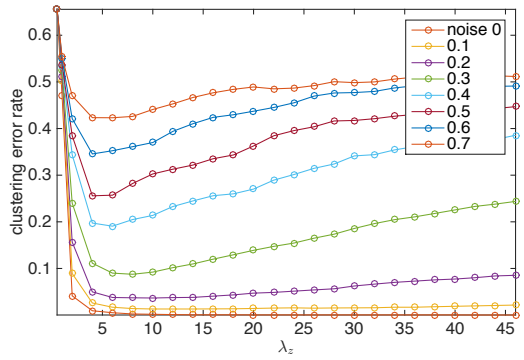
In the second experiment, we show that the optimal choice of parameter λ_z is not sensitive to the Gaussian noise level. The relative noise level varies from 0 to 0.7, and parameter λ_z varies from 10 to 250. The other settings are the same as the first experiment. According to Fig. 2, the performance of the proposed method is rather good for a large range of parameter λ_z. For different noise level, the best choice of parameter λ_z stays the same. The reference algorithm, however, is more sensitive to the parameter, the best choice of λ_z varies with the noise level. This validates our motivation that the proposed algorithm can be less sensitive to the noise energy.

There are no outliers in above experiments, as we set λ_e large to eliminate the effects of the term $\|\mathbf{E}\|_1$ in our proposed objective function. In the third experiment, we set λ_e to be smaller and consider the influence of the $\|\mathbf{E}\|_1$ term. We have 30 noise corrupted samples for each category and 10 outliers in total drawn from the unit sphere. We set $\lambda_e = 1.5$, and the other settings are the same as the second experiment. By optimizing the proposed objective function, we get the linear sparse representation matrix $\mathbf{C}$ and the outlier term $\mathbf{E}$ as shown in Fig. 3. The 10 outliers can be easily recognized using the energy of columns in $\mathbf{E}$, so we just label samples with energy larger than the average energy in $\mathbf{E}$ as outliers. Then we eliminate the columns and rows of matrix $\mathbf{C}$ corresponding to outliers and apply spectral clustering to the rest part. By doing this, we can discard all the 10 outliers at the cost of mistaking one or two normal sample.

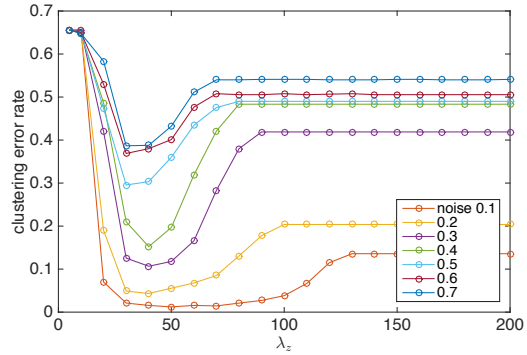
4.2. Real-world data

The proposed method is tested on a real sensor captured dataset, the Carnegie Mellon Motion Capture dataset [3]. There were sensor measurements at multiple joints of a human body captured at different time instances, and 149 subjects performing all kinds of activities were captured. We use the data of subject 86, consisting of 15 different trials, where each trial comprises multiple motion activities. Trials 2 and 5 are selected, because they have more motion activities, i.e., more categories, and are more challenging for the cluster algorithms. Samples in the dataset are vectors in 42-dimensional space. Each sample indicates a gesture of the captured human. Our purpose is to separate the sensory data into different activities, so that each cluster corresponds to a known activity, for example, jumping. Due to the biological constraints of human body, it is reasonable to assume that the sensory data lies in subspaces.

[3]available at `http://mocap.cs.cmu.edu`



(a) SSC



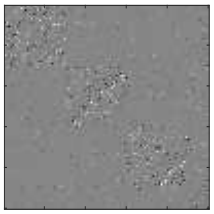
(b) SR-SSC

Fig. 2. The best choice of parameter λ_z in SR-SSC is less sensitive to the strength of noise.

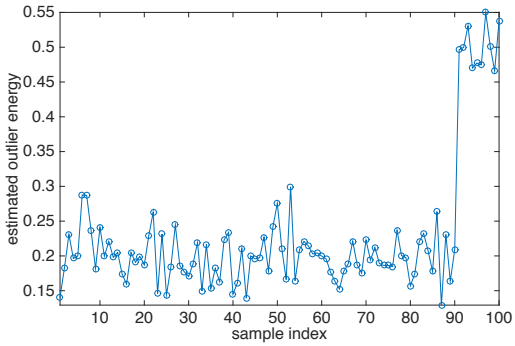
In the fourth experiment, we use three pairs of motion data for test, i.e., motion 1 and 2, motion 2 and 3, motion 3 and 5. When using algorithm SSC, we have to tune parameter λ_z for better result. However, as shown in Fig. 4, it is not possible to cluster all the test pairs using a single selection of λ_z. In contrast, when using our proposed SR-SSC, we can set $\lambda_z \approx 1$ to cluster all the three pairs of test motions, as shown in Fig. 4.

5. CONCLUSIONS

We study the sparse subspace clustering problem in presence of both sparse outliers and additive Gaussian noise. A convex optimization problem with a square-root penalty term is proposed to obtain a sparse self-representation, and a solving algorithm based on ADMM is devised. Solution to the outlier term $\mathbf{E}$ in our method points out outliers in the data, so we adopt a simple method to discard the outliers. The proposed model and algorithm are applied in motion segmentation. The clustering performances are demonstrated and analyzed on both synthetic data and real data. Numerical results show that our method is able to achieve higher cluster accuracy than the reference method both when the subspaces have intersections and not, and that the choice of the regularization parameter is less sensitive to the noise level in our method than that in the reference method.

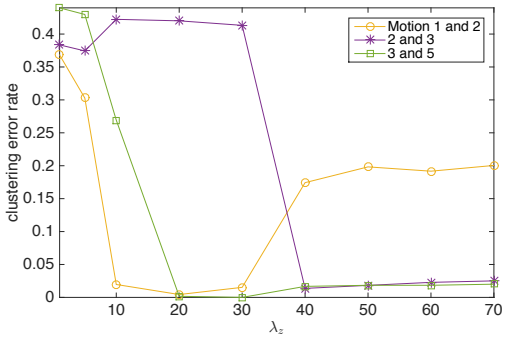


(a) sparse representation matrix $\mathbf{C}$

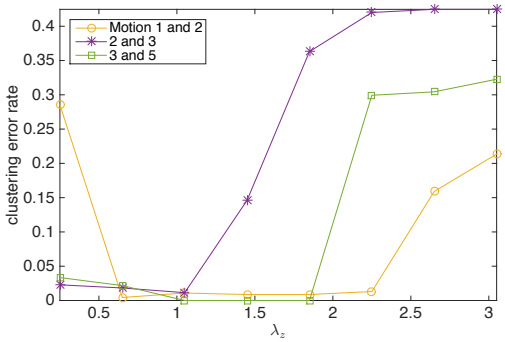


(b) estimated outlier energy

Fig. 3. Linear sparse representation matrix and outlier matrix estimated by solving (3). $\mathbf{C}$ clearly shows three categories and $\mathbf{E}$ shows the 10 outliers (the last 10 samples).



(b) SSC



(a) SR-SSC

Fig. 4. The best choice of parameter λ_z in SR-SSC is less sensitive to the strength of noise.

6. REFERENCES

[1] K. Kanatani, "Motion segmentation by subspace separation and model selection," in *IEEE International Conference on Computer Vision (ICCV)*, 2001, pp. 586–591 vol.2.

[2] A.Y. Yang, J. Wright, Y. Ma, and S. Shankar Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 212–225, 2008.

[3] W. Hong, J. Wright, K. Huang, and Y. Ma, "Multiscale hybrid linear models for lossy image representation.," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3655–71, 2006.

[4] B. Mcwilliams and G. Montana, "Subspace clustering of high-dimensional data: a predictive approach," *Data Mining and Knowledge Discovery*, vol. 28, no. 3, pp. 736–772, 2014.

[5] P. Indyk, "Approximate nearest neighbors: towards removing the curse of dimensionality," *Theory of Computing*, vol. 604-613, no. 11, pp. 604–613, 1998.

[6] J.B. Tenenbaum, V.D. Silva, and J.C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[7] D.L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[8] E.J. Candés, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *Journal of the Acm*, vol. 58, no. 3, 2009.

[9] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2790–2797.

[10] M. Soltanolkotabi, E. Elhamifar, and E.J. Candés, "Robust subspace clustering," *The Annals of Statistics*, vol. 42, no. 2, pp. 669–699, 2014.

[11] E.L. Dyer, A.C. Sankaranarayanan, and R.G. Baraniuk, "Greedy feature selection for subspace clustering," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2487–2517, 2013.

[12] R. Heckel, E. Agustsson, and H. Bölcskei, "Neighborhood selection for thresholding-based subspace clustering," in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2014, pp. 6761–6765.

[13] M. Soltanolkotabi, E. Elhamifar, and E.J. Candés, "Robust subspace clustering," *arXiv preprint arXiv:1301.2603*, 2013.

[14] 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.

[15] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[16] A. Belloni, V. Chernozhukov, and L. Wang, "Square-root lasso: Pivotal recovery of sparse signals via conic programming," *Biometrika*, vol. 98, no. 4, pp. 791–806, 2011.

[17] C. Thrampoulidis, A. Panahi, D. Guo, and B. Hassibi, "Precise error analysis of the lasso," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 3467–3471.

[18] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.