Deep Low-Rank Subspace Clustering

Mohsen Kheirandishfard Fariba Zohrizadeh Farhad Kamangar
Department of Computer Science and Engineering,
University of Texas at Arlington, USA
{mohsen.kheirandishfard,fariba.zohrizadeh}@gmail.com kamangar@cse.uta.edu

Abstract

This paper is concerned with developing a novel approach to tackle the problem of subspace clustering. The approach introduces a convolutional autoencoder-based architecture to generate low-rank representations (LRR) of input data which are proven to be very suitable for subspace clustering. We propose to insert a fully-connected linear layer and its transpose between the encoder and decoder to implicitly impose a rank constraint on the learned representations. We train this architecture by minimizing a standard deep subspace clustering loss function and then recover underlying subspaces by applying a variant of spectral clustering technique. Extensive experiments on benchmark datasets demonstrate that the proposed model can not only achieve very competitive clustering results using a relatively small network architecture but also can maintain its high level of performance across a wide range of LRRs. This implies that the model can be appropriately combined with the state-of-the-art subspace clustering architectures to produce more accurate results.

1. Introduction

Data clustering is the problem of partitioning a set of given sample points into multiple groups, called clusters so that the points within each cluster are more similar to each other than those in other clusters. Standard clustering algorithms mostly rely on specific distance measures (e.g. Euclidean distance) to perform clustering and compute the cluster memberships. However, these algorithms often fail to exhibit satisfactory performance in high-dimensional space due to some inherent structures and characteristics of the data (e.g. irregular patterns of data, curse of dimensionality, etc) (shown in Figure 1). Subspace clustering is a special data segmentation scenario which aims to partition a set of given points, drawn from a union of subspaces, into disjoint groups corresponding to the subspaces [3, 8, 26, 34]. There is a rich literature of works involving the cases with underlying linear subspaces [6, 9, 33, 34]. These studies often rely on the concept of self-expressiveness, stating that each sample point from a given dataset can be efficiently expressed as a linear (or affine) combination of other points [9]. Given that, the cluster memberships can be inferred using the assumption that the points of the same subspace can be expressed as a combination of each other. Motivated by this idea, various subspace clustering methods have been developed for different machine learning applications, such as motion segmentation [19, 32], face clustering [5, 45], movie recommendation [25, 44], etc. The basic idea behind these methods is to transform the input data into a new set of desired representations (e.g. sparse, low-rank) to build a graph showing the pairwise similarities between the entire points [9, 10, 20, 21, 35]. Then, a variant of the spectral clustering technique [24] is adopted to divide the graph into multiple sub-graphs corresponding to the underlying clusters. Among these methods, well-established sparse subspace clustering (SSC) [9] is a popular instance that adopts an ℓ_1-regularized model to limit the number of points used to reconstruct each sample point. This work has received special attention from the researchers in the field and many studies have been conducted to investigate its theoretical and practical aspects [30, 33, 42, 43].

Nowadays, deep models are increasingly used in various areas of clustering to boost the performance of traditional methods [7, 11]. In subspace clustering, deep models are adopted as powerful tools to exploit complex underlying patterns of data and learn suitable representations that satisfy the self-expressiveness property. This naturally leads to superior clustering performance compared to the conventional subspace clustering approaches [1, 15, 17, 29, 39, 49]. In particular, this superiority is more apparent in cases where the sample points reside on a union of non-linear subspaces since deep models can effectively capture the non-linearities of data and learn representations lying on a union of linear subspaces [15, 49]. A notable model is deep subspace clustering (DSC), introduced in [15], that utilizes a convolutional autoencoder with a self-expressive layer between the encoder and decoder to learn favorable deep subspace clustering representations.

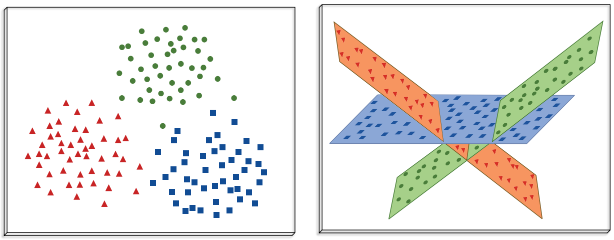
(a)
(b)

Figure 1: The distribution of points from two datasets. Samples are clustered together if (a) they are sufficiently close; (b) they belong to the same subspace. Distance-based clustering techniques are suitable when the points are uniformly distributed across the space. However, they perform poorly in cases where the data has a certain underlying structure.

In this work, we introduce a novel deep architecture that leverages convolutional autoencoders to learn low-rank representations (LRR) of the input data which are proven to be very suitable for the subspace clustering problem [35]. The proposed approach possesses the same architecture as the DSC method [15] except that we insert a fully connected linear layer and its transpose between the encoder and decoder to implicitly impose a rank constraint on the learned representations. Extensive experiments on benchmark datasets demonstrate the effectiveness of the proposed architecture compared to the DSC algorithm which is frequently adopted by various deep subspace clustering methods. In this sense, our model can serve as a proxy for the DSC model to enhance the performance of the state-of-the-art subspace clustering approaches [1, 17, 47, 49]. Moreover, we show that the proposed model not only requires much fewer learnable parameters compared to the DSC algorithm but also can maintain its high level of performance across a wide range of LRRs.

2. Related Works

A vast majority of subspace clustering approaches can be unified into a single framework consisting of two steps: 1) building a weighted graph with edges representing the similarity relationships between the points; 2) applying a variant of the spectral clustering technique [24] to partition the graph into multiple disjoint sub-graphs corresponding to different clusters [9, 12, 14, 31]. The following shows an optimization problem commonly used to obtain such a desired weighted graph:

$$\underset{\mathbf{C}\in\mathbb{R}^{n\times n}}{\text{minimize}} \quad \frac{1}{2}\|\mathbf{X}-\mathbf{X}\mathbf{C}\|_{\mathrm{F}}^{2}+\lambda\, g(\mathbf{C}) \qquad \text{(1a)}$$

$$\text{subject to} \quad \operatorname{diag}(\mathbf{C})=\mathbf{0}, \qquad \text{(1b)}$$

where $\mathbf{X} \in \mathbb{R}^{d\times n}$ is a data matrix with its columns denoting the sample points $\{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^n$, $\mathbf{C}$, called self-expression matrix, is a matrix whose $(i,j)^{th}$ element indicating the contribution of the i^{th} sample in reconstructing $\mathbf{x}_j$, $g : \mathbb{R}^{n\times n} \to \mathbb{R}$ is a specific regularization function, and $\lambda > 0$ is a fixed parameter to balance the contribution of different terms in (1a). The weighted graph can be constructed given the optimal solution of (1a)–(1b) and (1b) is an additional constraint to avoid getting the trivial solution $\overset{*}{\mathbf{C}} = \mathbf{I}_n$. To impose a desired structure on the graph, function $g(.)$ is mostly set to a specific matrix norm, such as $\|\mathbf{C}\|_0$ [40, 42], $\|\mathbf{C}\|_1$ [9], $\|\mathbf{C}\|_*$ [21, 35], $\|\mathbf{C}\|_{\mathrm{F}}$ [30], etc. Sparseness of $\mathbf{C}$ is a desirable property that is commonly enforced in the literature [9, 27]. Additionally, the low-rankness of matrix $\mathbf{C}$ well suits the problem of subspace clustering [2, 16, 27, 35]. Many approaches in the literature, such as robust principal component analysis [4], latent LRR [46], low-rank-sparse representation [36, 48], robust Kernel LRR [37], etc, leverage the low-rank representation of data to compute the clusters and recover the underlying subspaces.

Conventional subspace clustering approaches are mostly focused on the cases in which the points are drawn from linear subspaces. However, many applications are involved with sample points residing on a union of non-linear subspaces [15]. One empirical solution to deal with this non-linearity is to leverage kernel trick to implicitly map the data into a new space so that they better conform to linear subspaces [27, 28, 37, 41]. Unfortunately, this idea is not generally appealing as it is difficult to find an appropriate kernel for a given set of data points.

Recently, deep models are widely used in different areas of clustering to capture and encode the complex underlying pattern of the data [22, 23, 38]. Among them, autoencoders are frequently adopted by subspace clustering approaches to learn deep self-expressive representations from the input data. One pioneer approach that has received increasing attention in the literature is deep subspace clustering (DSC) [15]. This approach proposed to insert a fully connected linear layer between the encoder and decoder to generate suitable subspace clustering representations. The DSC model is further adopted by [49] to develop an adversarial approach for generating high-quality subspaces. [47] also used the DSC model to develop a novel end-to-end framework that achieves high-performance clustering results by jointly learning the representations, self-expression matrix, and the clustering results. More recently, [17] proposed an extension of the DSC by combining information from different levels of the encoder and decoder.

Notice that the aforementioned deep models mostly leverage $\|\mathbf{C}\|_1$ and $\|\mathbf{C}\|_2$ to learn desired representations. Despite the advantages derived from using LRR, deep models show less interest in using low-rank regularization con-

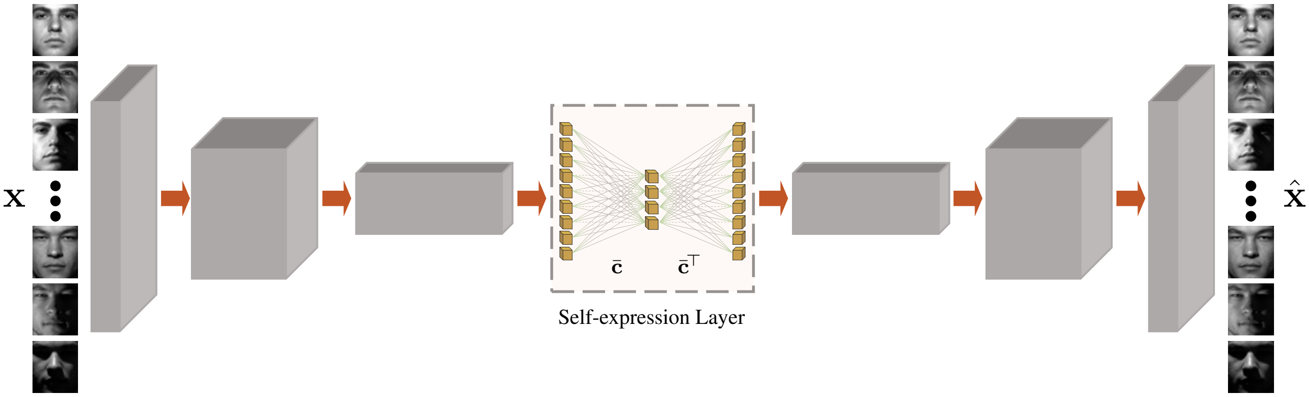

Figure 2: Deep Low-Rank Subspace Clustering Model: An example of the proposed architecture consisting of three convolutional layers in the encoder, three deconvolutional layers in the decoder, and two linear layers with weight matrices $\bar{\mathbf{c}}$ and $\bar{\mathbf{c}}^\top$. Observe that the self-expression layer can be defined as a linear layer whose weight matrix is $\mathbf{C} \triangleq \bar{\mathbf{c}}\bar{\mathbf{c}}^\top$. Since the rank of matrix $\bar{\mathbf{c}}$ is bounded by m ($m \ll n$), it is guaranteed that $\operatorname{rank}(\mathbf{C}) \leq m$ holds true. Such formulation allows to impose rank constraint on the self-expression matrix $\mathbf{C}$ by selecting different values of m.

straints (e.g. $\|\mathbf{C}\|_*$). The reason may stem from the fact that learning LRR is mostly involved with computing singular value decomposition (SVD) in each training step which is computationally cumbersome, especially when the problem dimension is large. Our approach introduces an idea to generate LRR without the requirement of computing SVD. We use the same architecture as the DSC algorithm except that we adopt a fully connected linear layer and its transpose to facilitate imposing a rank constraint on the self-expression matrix. Additionally, our model requires much fewer network parameters to learn compared to the DSC algorithm.

3. Problem Formulation

Compared to the conventional subspace clustering methods, deep models are extremely powerful to transform input data into a set of representations lying on a union of linear subspaces. The well-known DSC algorithm [15] develops a deep model consisting of an autoencoder with a fully-connected linear layer between the encoder and decoder to capture the non-linear relationships between data and produce self-expressive representations. Let $\check{\Theta}$, $\hat{\Theta}$, and $\mathbf{C}$ respectively show the parameters associated with the encoder, decoder, and the fully connected layer. The DSC algorithm aims to train the model by solving the following problem

$$\underset{\check{\Theta},\mathbf{C},\hat{\Theta}}{\text{minimize}} \quad \|\mathbf{X}-\hat{\mathbf{X}}\|_{\mathrm{F}}^2+\lambda\|\check{\mathbf{Z}}-\check{\mathbf{Z}}\mathbf{C}\|_{\mathrm{F}}^2+\gamma\|\mathbf{C}\|_p \qquad (2a)$$

$$\text{subject to} \quad \operatorname{diag}(\mathbf{C})=\mathbf{0}, \qquad (2b)$$

where $\mathbf{X}=[\mathbf{x}_1|\ldots|\mathbf{x}_n]\in\mathbb{R}^{d\times n}$ is a data matrix consisting of n d-dimensional sample points, $p\in\{1,2\}$ determines the norm type, $\mathbf{C}\in\mathbb{R}^{n\times n}$ shows the self-expression matrix, $\check{\mathbf{Z}}\in\mathbb{R}^{\check{d}\times n}$ denotes the encoder output where $\check{d}$ is the dimension of the latent space, and $\hat{\mathbf{X}}$, called reconstructed data matrix, shows the output of the decoder to the input matrix $\check{\mathbf{Z}}\mathbf{C}$. The main purpose of problem (2a) – (2b) is to capture the non-linearities of the input data and learn representations satisfying the self-expressiveness property (i.e. $\check{\mathbf{Z}}\approx\check{\mathbf{Z}}\mathbf{C}$). Given that, the spectral clustering technique can be applied to infer underlying subspaces and determine the sample assignments.

As noted earlier, LRR is proven to be very favorable for subspace clustering algorithms. This provides a strong motivation in leveraging powerful deep learning frameworks to generate deep LRR of data. In what follows, we propose a deep subspace clustering model that can implicitly impose a rank constraint on the self-expression matrix $\mathbf{C}$ to learn self-expressive LRRs.

4. Proposed Method

The most common technique to promote low-rankness on matrix $\mathbf{C}$ is to incorporate the nuclear norm regularization term $\|\mathbf{C}\|_*$ into the loss function. This technique might not be appealing for deep models due to the computational cost of computing its gradient in the backpropagation process. To deal with this issue and to learn low-rank representations, an alternative idea is to modify the DSC algorithm by considering an optimization problem of form

$$\underset{\check{\Theta},\mathbf{C},\hat{\Theta}}{\text{minimize}} \quad \|\mathbf{X}-\hat{\mathbf{X}}\|_{\mathrm{F}}^2+\lambda\|\check{\mathbf{Z}}-\check{\mathbf{Z}}\mathbf{C}\|_{\mathrm{F}}^2+\gamma\|\mathbf{C}\|_p \qquad (3a)$$

$$\text{subject to} \quad \operatorname{diag}(\mathbf{C})=\mathbf{0}, \qquad (3b)$$

$$\operatorname{rank}(\mathbf{C})\leq m, \qquad (3c)$$

where scalar m ($m \ll n$) is a hyperparameter that limits the maximum possible rank of matrix $\mathbf{C}$.

To design a deep subspace clustering model based on the formulation (3a) – (3c), we propose to replace the self-expression layer $\mathbf{C}$ in the model with a fully-connected linear layer $\bar{\mathbf{c}} \in \mathbb{R}^{n \times m}$ and its transpose. In this case, the self-expression layer can be seen as a symmetric matrix of form $\mathbf{C} \triangleq \bar{\mathbf{c}}\,\bar{\mathbf{c}}^\top$ where weight matrix $\bar{\mathbf{c}} \in \mathbb{R}^{n \times m}$ needs to be learned. With this definition and since the condition $\operatorname{rank}(\bar{\mathbf{c}}) \leq \min(m, n)$ is always valid, we can conclude that $\operatorname{rank}(\mathbf{C}) \leq m$ is guaranteed to hold true. Hence, it can be seen that the proposed architecture implicitly poses a rank constraint on the self-expression matrix $\mathbf{C}$.

Notice that the rank constraint (3c) avoids the trivial solution $\overset{*}{\mathbf{C}} = \mathbf{I}_n$ for (3a) – (3c) because $\operatorname{rank}(\bar{\mathbf{c}}\bar{\mathbf{c}}^\top)$ is guaranteed to not exceed m. Therefore, we can train the proposed architecture by solving the following minimization problem

$$\underset{\check{\Theta}, \bar{\mathbf{c}}, \hat{\Theta}}{\operatorname{minimize}} \ \|\mathbf{X} - \hat{\mathbf{X}}\|_{\mathrm{F}}^2 + \lambda \|\check{\mathbf{Z}} - \check{\mathbf{Z}}\bar{\mathbf{c}}\bar{\mathbf{c}}^\top\|_{\mathrm{F}}^2 + \gamma \|\bar{\mathbf{c}}\bar{\mathbf{c}}^\top\|_2. \quad (4a)$$

Problem (4a) can be solved using standard backpropagation technique. Given the optimal solution $\overset{*}{\bar{\mathbf{c}}}$ of the problem, we form the symmetric affinity matrix $\mathbf{W} = |\overset{*}{\bar{\mathbf{c}}}\,\overset{*}{\bar{\mathbf{c}}}^\top|$ representing the pairwise relationships between the sample points. Then, we apply a variant of the spectral clustering technique on matrix $\mathbf{W}$ to recover the underlying subspaces and determine the point assignments.

Figure 2 illustrates the proposed architecture in detail. The encoder output is fed into the self-expression layers consisting of a fully-connected linear layer and its transpose layer. The output of the self-expression layers is then used by the decoder to reconstruct the original data. Observe that the proposed model uses mn learnable parameters in the self-expression layer which is much fewer than that for the DSC algorithm which is n^2. This leads to a significant difference, particularly when the dataset size becomes large. Additionally, our approach facilitates incorporating nuclear norm regularization at a much lower computational cost due to the fact that $\|\bar{\mathbf{c}}\bar{\mathbf{c}}^\top\|_* = \|\bar{\mathbf{c}}^\top\bar{\mathbf{c}}\|_*$. Therefore, the regularization is involved with computing the nuclear norm of a $m \times m$ matrix which is computationally much cheaper than that for a $n \times n$ matrix since $m \ll n$.

In the following, we conduct multiple experiments on benchmark datasets to evaluate the performance of the proposed model on different subspace clustering tasks.

5. Experiments

In this section, we assess the performance of the proposed approach, named DLRSC, on three benchmark datasets: Extended Yale B, COIL20, and COIL100. Our results are compared against some baseline subspace clustering algorithms, such as Low Rank Representation (LRR) [20], Low Rank Subspace Clustering (LRSC) [35], Sparse Subspace Clustering (SSC) [9], SSC with the pre-trained convolutional auto-encoder features (AE+SSC), Kernel Sparse Subspace Clustering (KSSC) [28], SSC by Orthogonal Matching Pursuit (SSC-OMP) [42], Efficient Dense Subspace Clustering (EDSC) [13], EDSC with the pre-trained convolutional auto-encoder features (AE+EDSC), and Deep Subspace Clustering (DSC) [15].

Notice that it is not the intention of this work to compete with the state-of-the-art subspace clustering methods [17, 47] on different tasks. Instead, we aim to highlight the advantages offered by our architecture over the well-established DSC algorithm which is widely adopted by recent deep subspace clustering approaches in the literature [1, 17, 47, 49]. Moreover, we conduct additional experiments on the Extended Yale B dataset to evaluate the sensitivity of the DLRSC to the choice of parameter m.

In all experiments, we use the same network settings, pretraining and fine-tuning strategy as the DSC algorithm to provide a fair comparison. We adopted Adam optimizer [18] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and set the learning rate to 0.001 for training the network parameters. All implementations are done in PyTorch and the code will be publicly available on GitHub. In what follows, we provide hyperparameters used in each experiment separately and report the final results in terms of the clustering error, *i.e.* the percentage of the points that are incorrectly clustered:

$$\text{error} = \frac{\#\text{ of incorrectly clustered samples}}{\#\text{ of all samples}} \times 100\% \quad (5)$$

Extended Yale B: This is a human face dataset containing 2432 images of size 192 × 168 from 38 different subjects ($K = 38$), 64 images per each subject, where the images are taken under various illuminations and poses. Following the literature [9, 15], we down-sample the images to 48×42 for computational purposes. Motivated by the assumption that the underlying lower-dimensional subspaces share the same dimension, we set hyperparameter m to a number which is in proportion to the number of clusters. Hence, in our fine-tuning step, we set λ, γ, and m to 0.1, 1, and $10 \times K$ ($m \ll n = 64 \times K$), respectively, and adopt the standard backpropagation technique to train the model.

COIL20/COIL100: COIL100 is an object dataset consisting of 7200 images of size 32 × 32 from 100 different objects ($K = 100$), 72 images per each object, taken at pose intervals of 5 degrees. The hyperparameters used for conducting experiment on this dataset are as follows: $\lambda = 2$, $\gamma = 2$, $m = 10 \times K$; COIL20 is a smaller dataset consists of 1440 images of 20 different objects ($K = 20$) from the COIL100 dataset. We use $\lambda = 10$, $\gamma = 1$, and $m = 10 \times K$ ($m \ll n = 72 \times K$) for performing experiments on the COIL20 dataset.

Table 1: Clustering error (%) of different methods on Yale B, COIL20, and COIL100 datasets. The best results are in bold.

Dataset	LRR	LRSC	SSC	AE+SSC	KSSC	SSC-OMP	EDSC	AE+EDSC	DSC	DLRSC
Yale B	34.87	29.89	27.51	25.33	27.75	24.71	11.64	12.66	2.67	**2.47**
COIL20	30.21	31.25	14.83	22.08	24.65	29.86	14.86	14.79	5.14	**2.92**
COIL100	53.18	50.67	44.90	43.93	47.18	67.29	38.13	38.88	30.96	**28.14**

Table 2: Clustering error (%) of DLRSC method on Yale B for different choices of parameter m ($K = 38$).

m	$5 \times K$	$6 \times K$	$7 \times K$	$8 \times K$	$9 \times K$	$10 \times K$
error	4.65	3.24	3.66	4.12	2.75	2.47

The clustering error of the DLRSC algorithm on the aforementioned datasets are reported in Table 1. Observe that DLRSC outperforms the DSC algorithm and achieves very competitive results compared to the state-of-the-art methods by leveraging low-rankness property and learning representation lying on a union of linear subspaces. For each dataset, the proposed model only uses $n \times m$ learnable parameters in the self-expression layer which is much smaller than that for the DSC model which requires having $n \times n$ parameters. Besides these advantages, it should be noted that DLRSC replaces a single self-expression layer with two linear layers which in turn may increase the computational complexity, particularly for larger choices of m.

To evaluate the sensitivity of the DLRSC algorithm with respect to the rank of the self-expression matrix, we conduct multiple experiments on the Extended Yale B dataset for various choices of parameter m and report the corresponding clustering errors to Table 2. As it can be seen from the table, DLRSC can maintain the clustering error relatively small over a wide range of m which certifies the potential of the proposed model for handling low-rank subspace clustering problem.

6. Conclusions

This work proposed a novel deep subspace clustering approach that can efficiently transform input data into new representations lying on a union of linear subspaces. Motivated by advantages derived from using low-rank representation (LRR) of data, the proposed approach can effectively incorporate rank constraints into a deep learning framework to learn deep LRR. Our deep model can be seen as an extension of the well-established deep subspace clustering (DSC) algorithm that leverages low-rank representation to perform subspace clustering. Compared to the DSC model, the proposed model requires much fewer network parameters and allows to incorporate some rank-regularization terms (e.g. nuclear norm) at a very lower computational cost. Experiments demonstrate that our approach is very robust to the level of low-rankness and can achieve very competitive results on benchmark datasets. This highlights the potential of our model in promoting the performance of some recent deep subspace clustering approaches.

References

[1] Mahdi Abavisani and Vishal M Patel. Deep multimodal subspace clustering networks. *IEEE J. Sel. Topics Signal Process.*, 12(6):1601–1614, 2018. 1, 2, 4

[2] Mahdi Abavisani and Vishal M Patel. Multimodal sparse and low-rank subspace clustering. *Information Fusion*, 39:168–177, 2018. 2

[3] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. *Automatic subspace clustering of high dimensional data for data mining applications*, volume 27. ACM, 1998. 1

[4] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *J. ACM*, 58(3):1–37, 2011. 2

[5] Xiaochun Cao, Changqing Zhang, Huazhu Fu, Si Liu, and Hua Zhang. Diversity-induced multi-view subspace clustering. In *CVPR*, 2015. 1

[6] Guangliang Chen and Gilad Lerman. Spectral curvature clustering (scc). *Int. J. Comput. Vis.*, 81(3):317–330, 2009. 1

[7] Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with Gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016. 1

[8] Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *CVPR*, 2009. 1

[9] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11):2765–2781, 2013. 1, 2, 4

[10] Paolo Favaro, René Vidal, and Avinash Ravichandran. A closed form solution to robust subspace estimation and clustering. In *CVPR*, 2011. 1

[11] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *ICCV*, 2017. 1

[12] Reinhard Heckel and Helmut Bölcskei. Robust subspace clustering via thresholding. *IEEE Trans. Inf. Theory*, 61(11):6320–6342, 2015. 2

[13] Pan Ji, Mathieu Salzmann, and Hongdong Li. Efficient dense subspace clustering. In *WACV*, 2014. 4

[14] Pan Ji, Mathieu Salzmann, and Hongdong Li. Shape interaction matrix revisited and robustified: Efficient subspace clus-

tering with corrupted and incomplete data. In *ICCV*, 2015. 2

[15] Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian Reid. Deep subspace clustering networks. In *NeurIPS*, 2017. 1, 2, 3, 4

[16] Hao Jiang, Daniel P Robinson, René Vidal, and Chong You. A nonconvex formulation for low rank subspace clustering: algorithms and convergence analysis. *Comput. Optim. Appl.*, 70(2):395–418, 2018. 2

[17] Mohsen Kheirandishfard, Fariba Zohrizadeh, and Farhad Kamangar. Multi-level representation learning for deep subspace clustering. In *WACV*, 2020. 1, 2, 4

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 4

[19] Fabien Lauer and Christoph Schnörr. Spectral clustering of linear subspaces for motion segmentation. In *ICCV*, 2009. 1

[20] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):171–184, 2012. 1, 4

[21] Shirui Luo, Changqing Zhang, Wei Zhang, and Xiaochun Cao. Consistent and specific multi-view subspace clustering. In *AAAI*, 2018. 1, 2

[22] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *ICANN*, 2011. 2

[23] Sudipto Mukherjee, Himanshu Asnani, Eugene Lin, and Sreeram Kannan. ClusterGAN: Latent space clustering in generative adversarial networks. *arXiv preprint arXiv:1809.03627*, 2018. 2

[24] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *NeurIPS*, 2002. 1, 2

[25] Eirini Ntoutsi, Kostas Stefanidis, Katharina Rausch, and Hans-Peter Kriegel. Strength lies in differences: Diversifying friends for recommendations through subspace clustering. In *CIKM*, 2014. 1

[26] Lance Parsons, Ehtesham Haque, and Huan Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explor.*, 6(1):90–105, 2004. 1

[27] Vishal M Patel, Hien Van Nguyen, and René Vidal. Latent space sparse subspace clustering. In *ICCV*, 2013. 2

[28] Vishal M Patel and René Vidal. Kernel sparse subspace clustering. In *ICIP*, 2014. 2, 4

[29] Xi Peng, Shijie Xiao, Jiashi Feng, Wei-Yun Yau, and Zhang Yi. Deep subspace clustering with sparsity prior. In *IJCAI*, 2016. 1

[30] Xi Peng, Lei Zhang, and Zhang Yi. Scalable sparse subspace clustering. In *CVPR*, 2013. 1, 2

[31] Pulak Purkait, Tat-Jun Chin, Alireza Sadri, and David Suter. Clustering with hypergraphs: the case for large hyperedges. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(9):1697–1711, 2016. 2

[32] Shankar Rao, Roberto Tron, Rene Vidal, and Yi Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(10):1832–1845, 2010. 1

[33] Mahdi Soltanolkotabi, Emmanuel J Candes, et al. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, 40(4):2195–2238, 2012. 1

[34] René Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2011. 1

[35] René Vidal and Paolo Favaro. Low rank subspace clustering (LRSC). *Pattern Recognit. Lett.*, 43:47–61, 2014. 1, 2, 4

[36] Jun Wang, Daming Shi, Dansong Cheng, Yongqiang Zhang, and Junbin Gao. Lrsr: low-rank-sparse representation for subspace clustering. *Neurocomputing*, 214:1026–1037, 2016. 2

[37] Shijie Xiao, Mingkui Tan, Dong Xu, and Zhao Yang Dong. Robust kernel low-rank representation. *IEEE Trans. Neural Netw. Learn. Syst.*, 27(11):2268–2281, 2015. 2

[38] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, 2016. 2

[39] Xu Yang, Cheng Deng, Feng Zheng, Junchi Yan, and Wei Liu. Deep spectral clustering using dual autoencoder network. *arXiv preprint arXiv:1904.13113*, 2019. 1

[40] Yingzhen Yang, Jiashi Feng, Nebojsa Jojic, Jianchao Yang, and Thomas S Huang. ℓ^0-sparse subspace clustering. In *ECCV*, 2016. 2

[41] Ming Yin, Yi Guo, Junbin Gao, Zhaoshui He, and Shengli Xie. Kernel sparse subspace clustering on symmetric positive definite manifolds. In *CVPR*, 2016. 2

[42] Chong You, Daniel Robinson, and René Vidal. Scalable sparse subspace clustering by orthogonal matching pursuit. In *CVPR*, 2016. 1, 2, 4

[43] Chong You and Rene Vidal. Geometric conditions for subspace-sparse recovery. In *ICML*, 2015. 1

[44] Amy Zhang, Nadia Fawaz, Stratis Ioannidis, and Andrea Montanari. Guess who rated this movie: Identifying users through subspace clustering. *arXiv preprint arXiv:1208.1544*, 2012. 1

[45] Changqing Zhang, Huazhu Fu, Si Liu, Guangcan Liu, and Xiaochun Cao. Low-rank tensor constrained multiview subspace clustering. In *ICCV*, 2015. 1

[46] Hongyang Zhang, Zhouchen Lin, Chao Zhang, and Junbin Gao. Robust latent low rank representation for subspace clustering. *Neurocomputing*, 145:369–373, 2014. 2

[47] Junjian Zhang, Chun-Guang Li, Chong You, Xianbiao Qi, Honggang Zhang, Jun Guo, and Zhouchen Lin. Self-supervised convolutional subspace clustering network. In *CVPR*, 2019. 2, 4

[48] Li-Na Zhao, Fang Ma, and Hong-Wei Yang. Subspace clustering and feature extraction based on latent sparse low-rank representation. In *ICMLC*. IEEE, 2019. 2

[49] Pan Zhou, Yunqing Hou, and Jiashi Feng. Deep adversarial subspace clustering. In *CVPR*, 2018. 1, 2, 4