Overcomplete Deep Low-Rank Subspace Clustering

Yongmeng Feng
School of Computer Engineering
Jiangsu University of Technology
Changzhou Jiangsu China
fym199706@126.com

Congzhe You
School of Computer Engineering
Jiangsu University of Technology
Changzhou Jiangsu China
youcongzhe@126.com

***Abstract*—Aiming at the fact that when the input data in the deep subspace clustering networks (DSC) has noise, its robustness is poor, the performance is significantly degraded, and the method has too many learnable parameters, we suggest an overcomplete deep low-rank subspace clustering (ODLRSC). The technique is easy to use, efficient, and has shown to be a great fit for subspace clustering. By inserting a fully connected linear layer and its transposition between the encoder and decoder in our suggested technique, we may automatically put rank restrictions on the learnt representations. Additionally, in order to obtain a more reliable representation of the input data for clustering, the characteristics of the under- and over-complete auto-encoder networks are fused in the encoder. Our technique beats DSC and other clustering algorithms in the field of clustering error, and can sustain high level of quality throughout a broad range of LRRs, according to experimental findings on benchmark datasets.**

Keywords-deep subspace clustering; overcomplete representation; clustering error

I. INTRODUCTION

Subspace clustering algorithm is a common clustering algorithm for processing high-dimensional data. High-dimensional data is generally embedded in low-dimensional subspaces, and subspace clustering uses this feature to find reasonable clustering divisions in different subspaces. The objective of subspace clustering is really to figure out how many subspaces there are, how big they are, what each subspace's basis is, and how to divide the data given a dataset that results from the union of many subspaces. Subspace clustering techniques are often used in image segmentation, motion segmentation, picture clustering, movie recommendation, and several other computer vision applications because they perform well with high-dimensional data.

Deep models are being utilized more often as time goes on in several sectors of clustering to enhance the performance of conventional clustering techniques. In comparison to conventional subspace clustering techniques, this leads to greater clustering efficiency. As a result of deep models' ability to capture data nonlinearity and develop representations that reside on the union of linear subspaces, the benefit is particularly noticeable when sample points fall on the union of non-linear subspaces. The Deep Subspace Clustering Network (DSC), which he [1] introduced as the first deep learning-based approach, addresses the subspace clustering issue. With an intermediate self-expressive layer between the encoder and decoder, this network's undercomplete ("encoder-decoder") design makes use of a convolutional autoencoder to generate effective deep subspace clustering representations. There are two key issues with DSC, despite the fact that it performs well and much better than earlier techniques. The performance suffers greatly when there is data noise since the system is less resistant. The second is that the method has too many learnable parameters

In this essay, we put forward Overcomplete Deep Low-Rank Subspace Clustering Network (ODLRSC) as a new deep architecture. By using an excessively complex convolutional autoencoder, it results in overcomplete representation [2]. In order to implicitly provide rank restrictions on the learnt representation, a fully connected linear layer and its transpose are introduced between the encoder and decoder. And this architecture allows the convolutional autoencoder to be trained in parallel with the undercomplete autoencoder in DSC. We test the effectiveness of our technique using three statistical datasets: MNIST [3], COIL20 [4], and ORL [5]. Our tests demonstrate that ODLRSC is clearly superior than DSC and other conventional subspace clustering techniques. This architecture not only has better robustness [6], but also requires fewer learnable parameters.

II. RELATED WORK

Subspace clustering was initially resolved using linear techniques: first, a matrix of affinity was created by calculating the affinity between every matched set of data points, and then, using various spectral clustering techniques [7], different clusters were identified by applying the techniques to the affinity matrix. This approach may be represented mathematically as the following optimisation issue:

$$\min_{C}\|C\|_p + \frac{\lambda}{2}\|\mathrm{X} - \mathrm{X}C\|_F^2 \quad s.t.\ (diag(C) = 0) \tag{1}$$

Where $\mathrm{X} \in \mathrm{R}^{d\times n}$ is a data matrix whose columns represent sample points $x_i \in \mathrm{R}^{d\times n}$, C is a self-expression matrix, p=0 or 1 or core bound norm. Here, the diagonal restriction prevents finding easy solutions. However, this clustering technique is only appropriate for the subspaces of linear. If the data points are not in the subspace of linear, one approach is to implicitly translate the data to a new space using the kernel method [8], which will improve how well the data fits the linear subspace. However, this method makes it challenging to choose a good kernel for a specific collection of data points.

Subspace clustering techniques often employ autoencoders to extract representations of one's deeper self-derived from one's incoming data as deep models become more and more prevalent in computer vision and machine learning applications. Among them, Deep Subspace Clustering (DSC) introduces a new self-expression layer for deep autoencoders, that is, between the encoder and the decoder is a completely linked linear layer that has been introduced, which facilitates the generation of befitting subspace clustering representations.

2473-3636/22/$31.00 ©2022 IEEE
DOI 10.1109/DCABES57229.2022.00048

In [9], the DSC model was further utilized in order to build adversarial approaches for the generation of superior quality subspaces. This work was based on DSC. Another end-to-end framework, based on the DSC model, has been created by [10] that delivers high-performance outcomes in the clustering process via collaborative education, self-expression matrices, and clustering results. Unlike the above methods, in the encoder, we make use of an overcomplete representation, and for the purpose of making it easier to impose rank restrictions on the self-expressing matrix, we make use of a fully linked linear layer in addition to its transpose. In comparison to the DSC technique, our model is more resilient and needs the learning of a much less number of network parameters.

III. Overcomplete Deep Low-Rank Subspace Clustering

In order to increase the performance of clustering, our proposed technique takes use of overcomplete representations as well as low-rank representations [11]. In this part, we will first provide a concise overview of our suggested network architecture, and then proceed to detail our refined approach.

A. Network Structure

We use two encoders trained concurrently in ODLRSC. One has an up-sampling layer after each convolutional layer, while the other has a max-pooling layer. We provide a generalized method to overcomplete signal representation. It entails employing an overcomplete basis, which increases the number of basic functions beyond the number of samples of the input signal. As a result, the overcomplete representation is proved to be more stable and has better flexibility in capturing the structure of the data. For simplicity, we use bilinear interpolation for up-sampling in the network architecture.

The completely linked linear layer and its transposed layer make up the layer for self-expression [12], and their weights match up with the coefficients of the self-expressive representation matrix C. The affinity matrix is immediately learned by this layer. Convolutional and up-sampling layers make up the decoder. The decoder reconstructs the original data using the self-expression layer's output. The self-expression layer of this model employs m × n learnable parameters, which is much less than the n^2 of the DSC method and results in a considerable difference. In order to produce clusters, spectral clustering is performed with the learnt affinity matrix as the input. The enhanced network structure and ODLRSC strategy are shown in Figure 1.

B. Improved Method

In the beginning of our training, our autoencoder is trained independently utilizing reconstruction loss [13]. The formula for reconstruction loss L_r is as follows:

$$L_r = \|x - \hat{x}\|_F^2 \tag{2}$$

Where $X = [X_1|\dots|X_n] \in R^{d\times n}$ is a data matrix that consists of n sample points that have d dimensions each, $\hat{X}$ is the decoder-to-input matrix reconstruction.

When training the rebuilt network, we make use of the Adam's 0.001 learning rate across all of our trials. The pretrained weights are first loaded into the network, and then the network is fine-tuned using a layer of self-expression and a loss term of self-expression L_{self} with the following formula:

$$L_{self}(\theta, C) = \lambda_2\|C\|_p + \frac{\lambda_3}{2}\left\|Z_{\theta_e} - Z_{\theta_e}C\right\|_F^2 \tag{3}$$

In the equation, θ stands for the network's parameters, and θ_e stands for the encoder's parameters. Z is the matrix that reflects the latent representation that is present in the network's self-expression layer, and C is the matrix that reflects the coefficients of self-representation. In the fine-tuning step, we train the network using L2 regularization on C (p=2).

When it comes to optimizing combined losses from these two events, we rely on the Adam optimizer. The formula for the overall loss, L_{Total}, is as follows:

$$L_{Total} = \frac{\lambda_1}{2}\left\|X - \hat{X}\right\|_F^2 + \lambda_2\|C\|_p + \frac{\lambda_3}{2}\left\|Z_{\theta_e} - Z_{\theta_e}C\right\|_F^2 \tag{4}$$

Where the hyperparameters λ_1, λ_2, and λ_3 regulate the degree to which each individual loss term affects the ultimate loss. Different datasets have different hyperparameter settings. The following section will go through the particular adjustments to be made to the hyperparameters.

When a low-rank constraint is added to the self-expression layer, the following optimization issue may be solved using the following formula to train the model.

$$\min_C \lambda_2\|C\|_p + \frac{\lambda_1}{2}\left\|X - \hat{X}\right\|_F^2 + \frac{\lambda_3}{2}\left\|Z_{\theta_e} - Z_{\theta_e}C\right\|_F^2 \tag{5}$$

$$\text{subject to} \quad diag(C) = 0 \tag{6}$$

$$rank(C) \le m \tag{7}$$

In the above formula, a hyperparameter known as the scalar $m(m \ll n)$ places a cap on the highest potential rank that may be achieved by matrix C.

The self-expressive layer C of the network design is replaced with a completely linked linear layer $\bar{c} \in R^{n\times m}$ and its transpose. At this time, we can regard the self-expressive layer as a symmetric matrix of $C \triangleq \bar{c}\,\bar{c}^T$, where the weight matrix $\bar{c} \in R^{n\times m}$ need to learn, according to this definition, we can know that the $rank(C) \le m$ always holds. As a consequence of this, the self-expressive matrix C is subject to an implicit rank limitation imposed by our network design.

As a result, we are able to drill the suggested network topology by finding the optimal solution to the following minimization issue.

$$\min_{\bar{c}} \lambda_2\left\|\bar{c}\,\bar{c}^T\right\|_p + \frac{\lambda_1}{2}\left\|X - \hat{X}\right\|_F^2 + \frac{\lambda_3}{2}\left\|Z_{\theta_e} - Z_{\theta_e}\bar{c}\,\bar{c}^T\right\|_F^2 \tag{8}$$

Standard backpropagation methods [14] may be used to tackle this minimization issue. Given that the issue can only have one correct answer, $\bar{c}_l$, we construct a balanced affinity matrix [15] $W = \left|\bar{c}_l\bar{c}_l^T\right|$, which is used to represent the pairwise relationship of sample points. We next retrieve the fundamental subspace and ascertain the assignment using spectral clustering techniques on matrix W.

The model's performance on various subspace clustering tasks is then evaluated via a series of tests using benchmark datasets.

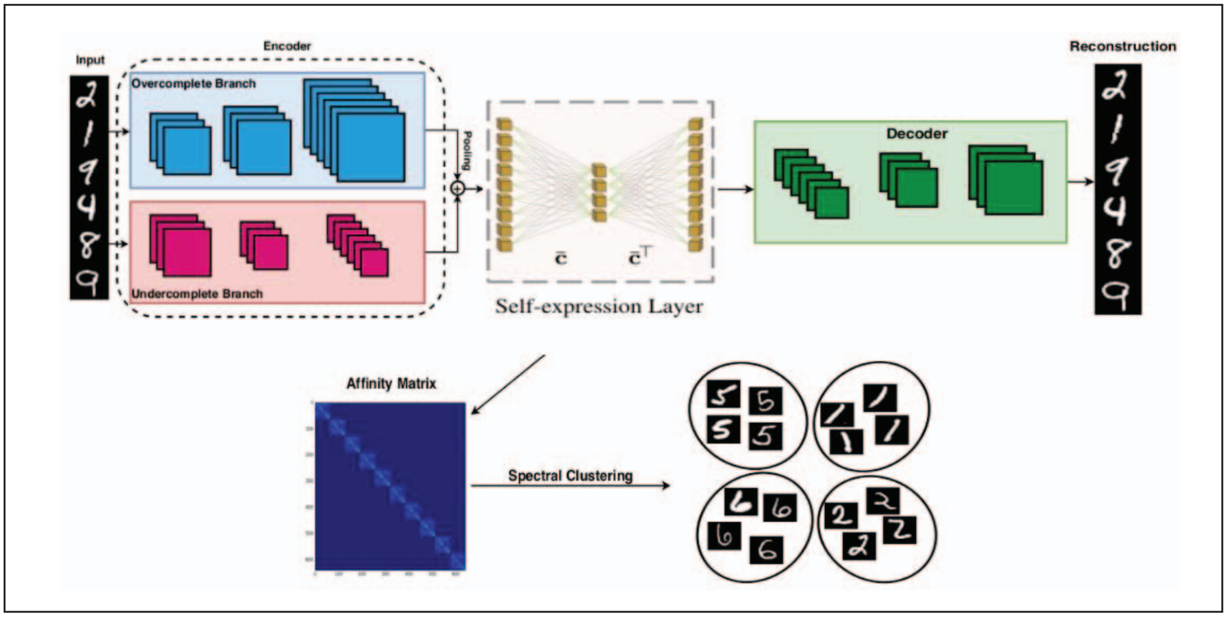

Figure1. Generall strategy for the suggested ODLRSC technique.

IV. EXPERIMENTS

We use TensorFlow 1.14 in PyCharm to conduct all of our experiments, and then we use three datasets—MNIST, COIL20 and ORL—to assess our methodology. Our method is compared with low-rank subspace clustering (LRSC) [15], sparse subspace clustering (SSC) [16], low-rank representation (LRR) [17], Efficient Dense Subspace Clustering (EDSC) [18], Kernel Sparse Subspace Clustering (KSSC) [19], and Deep Subspace Clustering (DSC). We go into depth about the hyperparameters we utilize for each dataset in the sections that follow, which differs in each dataset due to the different amount of data. In all of our quantitative analyses, the clustering error rate is something we employ, which may be described as follows:

$$error = \frac{\neq \text{ of wrongly clustered samples}}{\neq \text{ of all samples}} \times 100\% \tag{9}$$

A. Mnist Dataset

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. The handwritten digit pictures in the MNIST dataset are organized into 10 categories, ranging from 0 to 9 (K=10). We make a selection of 100 photographs at random from each category and perform subspace clustering using the set of these 1000 images with an image size of 28 × 28. In our network design (ODLRSC) for this dataset, the overcomplete encoder has two convolution blocks, the undercomplete encoder has three convolution blocks, and the decoder has three convolution blocks. In the encoder and decoder, the convolutional layers that kernel size is 55 for the very topmost layer and 33 for each subsequent layer. There are 20 and 10 filters in the overcomplete encoder, respectively. And there are 20, 10 and 5 filters in the overcomplete decoder, respectively. When it comes to the decoder, every convolutional block has a different number of filters: 5, 10, and 20, respectively. During the fine-tuning phase, we established the following parameters: λ_1=1.00, λ_2=10.00, λ_3=0.2 and m=8×K (m≪ n=64×K).

B. Coil20 Dataset

The equations are an exception to the prescribed specifications of this template. COIL20 is an image dataset that has a total of 1440 photos with a resolution of 32 x 32 pixels, 72 photographs for each of the 20 unique images(K=20) that make up the dataset, and many samples of the same images were photographed from a variety of angles. For this dataset, our network design (ODLRSC) contains one convolution block in the decoder and one convolution block in each of the overcomplete and undercomplete encoders. The encoder and decoder's convolutional layers have a kernel size of 3 by 3 and 15 filters in each convolutional layer. During the fine-tuning phase, we established the following parameters: λ1=1.00, λ2=1.00, λ3=12.00 and m=8×K (m≪ n=72×K).

C. Orl Dataset

ORL is a face image dataset including 400 face pictures of size 32 by 32 from 40 distinct persons(k=40), each with 10 samples. The ORL dataset is comprised of photos with a variety of face expressions captured under a variety of lighting conditions. In our network structure (ODLRSC) for this dataset, the overcomplete encoder has two convolution blocks, the undercomplete encoder has three convolution blocks, and the decoder has three convolution blocks. In both the encoder and the decoder, all of the convolutional layers have a core size of 3×3. The number of filters in the encoder is 3, 3 and 6, while in the decoder it is the opposite. During the fine-tuning phase, we established the following parameters: λ_1=1.00, λ_2=4.00, λ_3=0.2 and m=2×K (m≪ n=10×K).

D. Result Analysis

The clustering errors for datasets MNIST, COIL20 and ORL are shown in Tables 1, 2 and 3. In the following three tables, our experimental results are expressed in the

Table 1. Errors in clustering for the MNIST dataset compared to current approaches.

Means	LRSC	SSC	LRR	EDSC	KSSC	DSC	ODLRSC
Error	48.57 ± 0.18	54.67 ± 0.23	46.14 ± 0.19	43.51 ± 0.17	47.78 ± 0.18	24.97 ± 0.16	19.94 ± 0.14

Table 2. Errors in clustering for the COIL20 dataset compared to current approaches.

Means	LRSC	SSC	LRR	EDSC	KSSC	DSC	ODLRSC
Error	31.43 ± 0.17	14.78 ± 0.14	30.58 ± 0.18	14.76 ± 0.19	24.63 ± 0.16	5.13 ± 0.13	3.49 ± 0.11

Table 3. Errors in clustering for the ORL dataset compared to current approaches.

Means	LRSC	SSC	LRR	EDSC	KSSC	DSC	ODLRSC
Error	33.57 ± 0.16	32.54 ± 0.21	38.27 ± 0.19	27.26 ± 0.17	34.26 ± 0.17	14.07 ± 0.15	13.19 ± 0.13

form of adding and subtracting standard deviations from the mean of multiple experiments. From these three tables, we can see that the performance of the DSC method has been significantly improved after adding the deep learning method. The performance of our proposed ODLRSC method has been improved on the basis of DSC. Therefore, our suggested approach outperforms the DSC approach. This benefits from our improvement on the DSC algorithm, by exploiting overcomplete representation and low rank representations. In comparison to the most recent and cutting-edge techniques, the ODLRSC produces respectable outcomes.

V. Conclusion

In this paper, we put forward an overcomplete deep subspace clustering method based on low-rank representations, namely ODLRSC. Using a combination of overcomplete and undercomplete networks, inserting fully connected linear layers and their transposes between the encoder and decoder, utilizes both overcomplete and low-rank representations to perform subspace clustering. Compared with the DSC method, the model has better robustness and requires fewer network parameters. The outcomes of our suggested ODLRSC approach on benchmark datasets are satisfactory, according to experiments.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 61902160, 61806088，62002142), the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant No.19KJB520006), Natural Science Foundation of Jiangsu Province (No. BK20201057) and Jiangsu Province University-level Postgraduate Practice Innovation Plan Project (Grant No. XSJCX21_15).

References

[1] Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian Reid. Deep subspace clustering networks. In Advances in Neural Information Processing Systems, pages 24–33, 2017.

[2] Michael S Lewicki and Terrence J Sejnowski. Learning over-complete representations. Neural computation, 12(2):337–365, 2000.

[3] Yann LeCun, Léon Bottou, Y oshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[4] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-20). 1996.

[5] Ferdinando S Samaria and Andy C Harter. Parameterisation of a stochastic model for human face identification. In Proceedings of 1994 IEEE workshop on applications of computer vision, pages 138–142. IEEE, 1994.

[6] Xianzhen Li, Zhao Zhang, Yang Wang, Guangcan Liu, Shuicheng Yan, and Meng Wang. Multilayer collaborative low-rank coding network for robust deep subspace discovery. arXiv preprint arXiv:1912.06450, 2019.

[7] Andrew Y Ng, Michael I Jordan, and Y air Weiss. On spectral clustering: Analysis and an algorithm. In NeurIPS, 2002.

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

[9] Pan Zhou, Y unqing Hou, and Jiashi Feng. Deep adversarial subspace clustering. In CVPR, 2018.

[10] 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.

[11] V . M. Patel, H. V an Nguyen, and R. Vidal. Latent space sparse and low-rank subspace clustering. IEEE Journal of Selected Topics in Signal Processing, 9(4):691–701, 2015.

[12] M. Abavisani and V . M. Patel. Domain adaptive subspace clustering. In British Machine Vision Conference, 2016.

[13] Mohsen Kheirandishfard, Fariba Zohrizadeh, and Farhad Kamangar. Multi-level representation learning for deep subspace clustering. In WACV, 2020.

[14] Xu Yang, Cheng Deng, Feng Zheng, Junchi Y an, and Wei Liu. Deep spectral clustering using dual autoencoder network. arXiv preprint arXiv:1904.13113, 2019.

[15] René Vidal and Paolo Favaro. Low rank subspace clustering (lrsc). Pattern Recognition Letters, 43:47–61, 2014.

[16] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and applications. IEEE transactions on pattern analysis and machine intelligence, 35(11):2765–2781, 2013.

[17] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. IEEE transactions on pattern analysis and machine intelligence, 35(1):171–184, 2012.

[18] Pan Ji, Mathieu Salzmann, and Hongdong Li. Efficient dense subspace clustering. In IEEE Winter Conference on Applica-tionsof Computer Vision, pages 461–468. IEEE, 2014.

[19] Vishal M Patel and René Vidal. Kernel sparse subspace clus-tering. In 2014 ieee international conference on image pro-cessing (icip), pages 2849–2853. IEEE, 2014.