When to Learn What: Deep Cognitive Subspace Clustering

Yangbangyan Jiang[1,2], Zhiyang Yang[1,2], Qianqian Xu[3], Xiaochun Cao[1,2], Qingming Huang[3,4,5*]

中国科学院信息工程研究所 INSTITUTE OF INFORMATION ENGINEERING,CAS

中国科学院大学 University of Chinese Academy of Sciences

1. SKLOIS, IIE, CAS. 2. Sch. Cyber Security, UCAS. 3. KLIIP, ICT, CAS.

4. Sch. Comupter and Control Engineering, UCAS. 5. KLBDMKM, CAS

{jiangyangbangyan, yangzhiyong caoxiaochun}@iie.ac.cn, xuqianqian@ict.ac.cn, qmhuang@ucas.ac.cn

ICT 中国科学院计算技术研究所 INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

Introduction

Subspace clustering aims at clustering data points drawn from a union of low-dimensional subspaces. Recently deep neural networks are introduced to improve the performance. However, such models are sensitive to noise and outliers, since both difficult and easy samples are treated equally. On the contrary, in the human cognitive process, individuals tend to follow a learning paradigm from *easy to hard* and *less to more*. Inspired by such learning scheme, in this paper, we propose a robust deep subspace clustering framework, Deep Cognitive Subspace Clustering (**DeepCogSC**).

Model Formulation

- **Deep Subspace Clustering**: As depicted in Figure 1, we integrate a self-expressive layer into a deep auto-encoder. Let Θ_E, Θ_D, Θ_S be the parameters of the encoder, decoder and self-expressive layer, respectively.

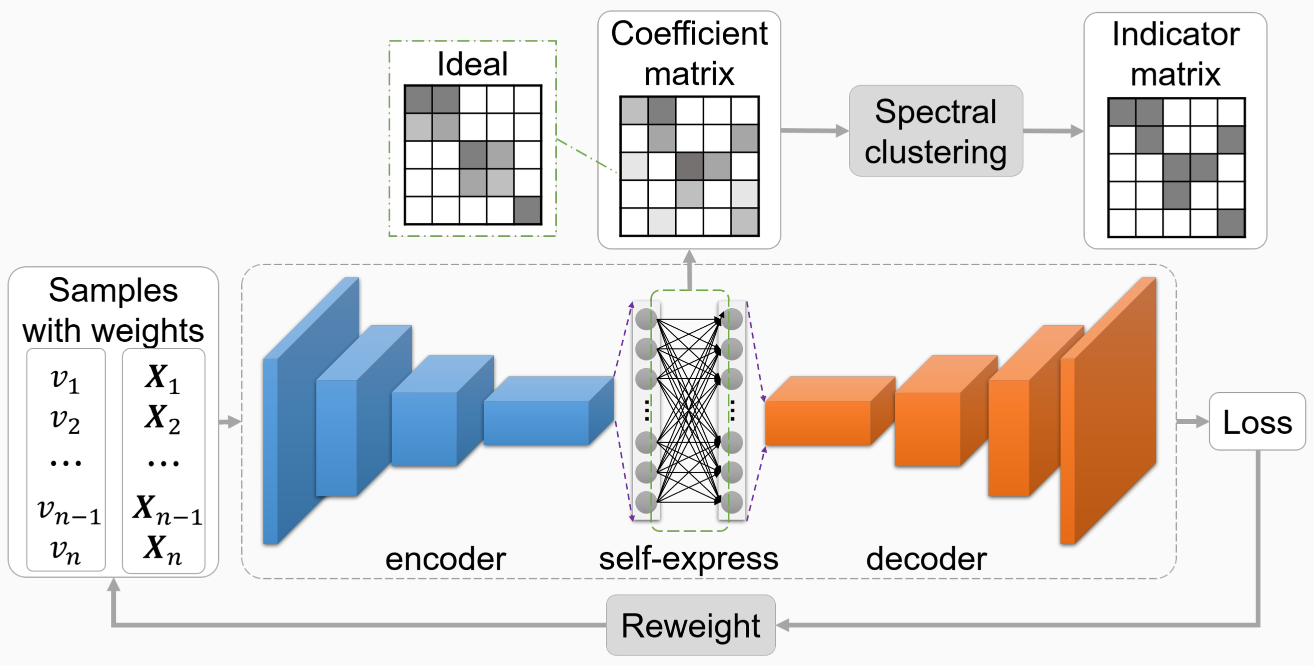

Fig. 1: Framework of Deep Cognitive Subspace Clustering.

→ **Data Reconstruction with Θ_E and Θ_D.** The deep conv auto-encoder transforms non-linear data into a linear latent space and reconstructs them.

→ **Latent Self-expressiveness with Θ_S.** A fully-connected layer without bias and activation function is placed after the encoder to "self-express" the data. A feature matrix Z is fed into this layer, formed by the flatten vector z_i of each feature map. $\Theta_S \in \mathbb{R}^{n\times n}$ is the coefficient matrix for $Z = Z\Theta_S$.

Denote whole parameter set as $\Theta = \{\Theta_E, \Theta_D, \Theta_S\}$, we have

$$\ell_i(\Theta) = \frac{1}{2}\left\|X_i - \hat{X}_i\right\|_F^2 + \frac{\lambda_1}{2}\left\|z_i - Z\Theta_{S,i}\right\|_2^2 \quad (1)$$

- **Self-pacing**: Based on the clustering network, we weight the losses by $\boldsymbol{v}$ to adjust the contribution of samples to the final objective:

$$\mathcal{L}_i(\Theta, \zeta) = v_i \ell_i(\Theta) + f(v_i, \zeta) \quad (2)$$

where ζ is to controll the learning pace and $f(v_i, \zeta)$ is the self-pace regularizer.

- **Overall**: With a structure regularizer $\Omega(\Theta_S) = \|\Theta_S\|_p$, we eventually reach our objective:

$$\begin{aligned} E(\boldsymbol{v}, \Theta, \zeta) &= \sum_{i=1}^{n} \mathcal{L}_i(\Theta, \zeta) + \lambda_2 \Omega(\Theta_S) \\ &= \sum_{i=1}^{n}\left[v_i\left(\frac{1}{2}\left\|X_i - \hat{X}_i\right\|_F^2 + \frac{\lambda_1}{2}\left\|z_i - Z\Theta_{S,i}\right\|_2^2\right) + f(v_i,\zeta)\right] + \lambda_2\left\|\Theta_S\right\|_p \end{aligned} \quad (3)$$

We specify $E(\boldsymbol{v}, \Theta, \zeta)$ via defining a novel regularizer in the following:

$$f(v, \zeta) = \zeta(v \log v - v) \quad (4)$$

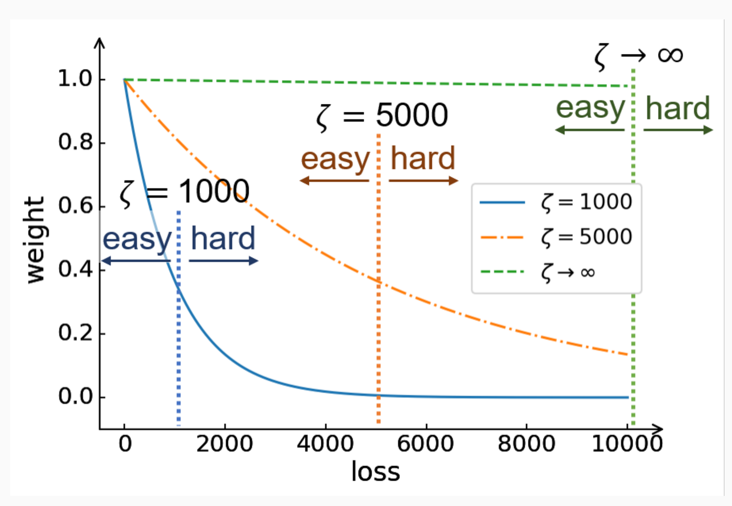

Fig. 2: The change of weight v with respect to the age parameter ζ and loss ℓ.

Optimization

- **Update $\boldsymbol{v}$**: We fix Θ at Θ^k. Since E is convex w.r.t. v_i, v_i's closed-form optimal solution is:

$$v_i^*(\Theta^k, \zeta) = e^{-\frac{\ell_i(\Theta^k)}{\zeta}} \quad (5)$$

→ $v^* \propto 1/\ell$: Easy samples are often preferred because of their smaller losses.

→ $v^* \propto \zeta$: As learning goes on, more hard samples are included into training.

- **Update ζ**: We predefine an ascending sequence $N = \{N_t\}_{t=1}^m$ to explicitly set # of selected samples in t-th stage. To keep $\zeta \propto t$, we sort the losses in ascending order and use $\tau \in (0, 1]$ to set:

$$\zeta^{t+1} = \max\{\tilde{\ell}_{N_t}, (1+\tau)\zeta^t\} \quad (6)$$

- **Update Θ**: Fixing v_i at v_i^*, we can update Θ through back-propagation.

Algorithm 1 Self-paced deep subspace clustering algorithm

Input: Dataset $\mathcal{D} = \{X_i\}_{i=1}^n$
Output: Parameters Θ, clustering labels $\boldsymbol{y} = [y_1, \cdots, y_n]$

```
Initialize Θ,  N ← {N_1, ..., N_m = n},  ζ ← 0
for t ← 1 to m do
    Update ζ with Eq. (6)
    while not converged do
        Update each v with Eq.(5)                        ▷ Fix Θ
        while not converged do
            Update Θ with a gradient-based solver        ▷ Fix v
        end while
    end while
end for
Z ← |Θ_S| + |Θ_S^T|
y = SpectralClustering(Z)
```

Theoretical Analysis

Proposition 1 (Implicit Objective Function). *With fixed ζ, the alternative optimization strategy for minimizing Eq. (3) is equivalent to the majorization-minimization algorithm for solving $\sum_{i=1}^n F_\zeta(\ell_i(\Theta)) + \lambda_2\left\|\Theta_S\right\|_p$, where*

$$F_\zeta(\ell) = \int_0^\ell v^*(l, \zeta)\mathrm{d}l = \zeta(1 - e^{-\frac{\ell}{\zeta}}) \quad (7)$$

Proposition 2 (Robustness). *Suppose that $\min_k \ell_k > B$ and $B < \infty$, we have, for any pair of distinct instances (i,j) in training dataset $\mathcal{D}$:*

$$\left|F_\zeta(\ell_i) - F_\zeta(\ell_j)\right| \le e^{-\frac{B}{\zeta}} \cdot \left|\ell_i - \ell_j\right|$$

With the Taylor expansion of $F_\zeta(\cdot)$, we have:

$$F_\zeta(\ell) = \zeta\left(1 - \sum_{n=0}^{\infty}\frac{\left(-\frac{\ell}{\zeta}\right)^n}{n!}\right) = \ell - o(\frac{1}{\zeta}) \quad (8)$$

Since $\lim_{\zeta\to\infty} F_\zeta(\ell) = \ell$, $F_\zeta(\ell)$ degenerates to the original ℓ after sufficient steps of iterations.

Experiments

Table 1: Clustering error rate (%) on Extended Yale B.

	10 subjects	20 subjects	30 subjects	all
LRR	22.22	30.23	37.98	34.87
LRSC	30.95	28.76	30.64	29.89
SSC	10.22	19.75	28.76	27.51
AE+SSC	17.06	18.23	19.99	25.33
KSSC	14.49	16.55	20.49	27.75
SSC-OMP	12.08	15.16	20.75	24.71
EDSC	5.64	9.30	11.24	11.64
AE+EDSC	5.46	7.67	11.56	12.66
DSC-Net-L1	2.23	2.17	2.63	3.33
DSC-Net-L2	**1.59**	**1.73**	2.07	2.67
DeepCogSC-L1 (Ours)	1.89	1.96	**1.93**	**2.38**
DeepCogSC-L2 (Ours)	**1.46**	**1.54**	**1.83**	**2.18**

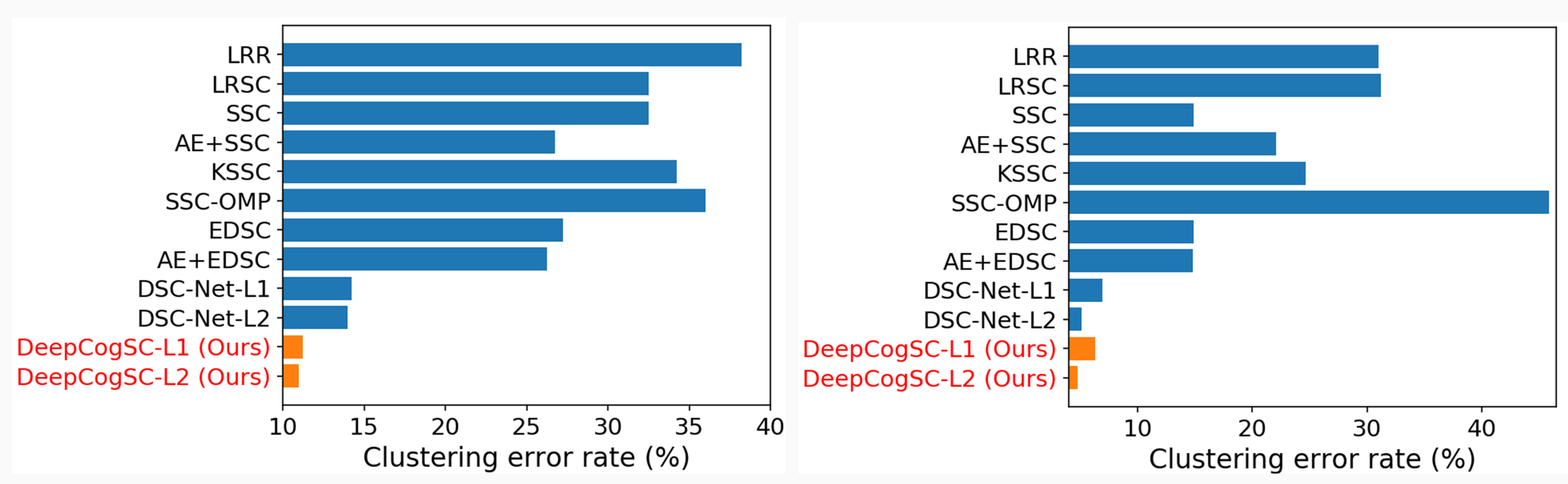

Fig. 3: Clustering error rate (%) on ORL and COIL20.

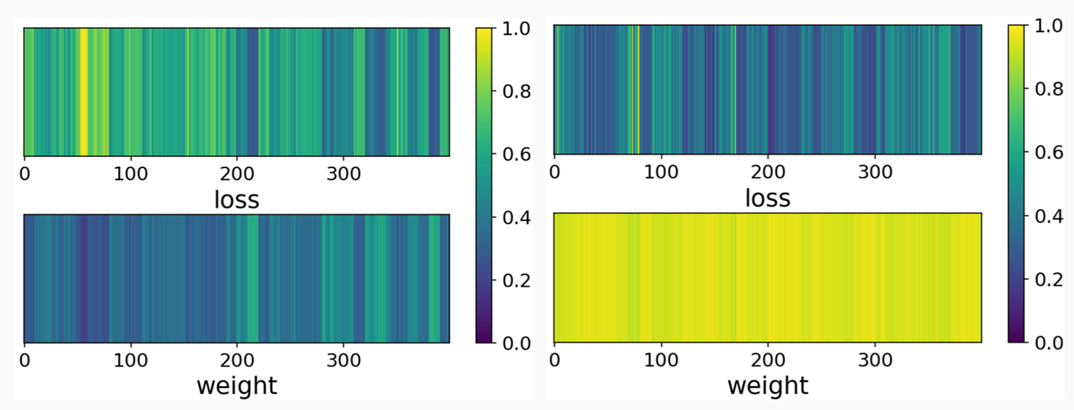

Fig. 4: Visualization of the weights and losses at 1st and 4th iteration. The losses are normalized.

Conclusion

We introduce a novel deep subspace clustering method called DeepCogSC. It takes sample difficulty into consideration, where subspace clustering is performed in a self-paced manner based on a deep auto-encoder. Subsequently, we propose an alternative optimization strategy to solve the problem. Theoretical analysis shows that DeepCogSC is more robust toward hard examples and outliers. Moreover, DeepCogSC tends to be consistent with the original subspace clustering algorithm after sufficient steps of iterations. Furthermore, experimental results demonstrate the superiority of DeepCogSC.