Proceedings of Machine Learning Research 101:1001–1016, 2019 ACML 2019

Self-Supervised Deep Multi-View Subspace Clustering

Xiukun Sun SUNXIUKUN@BJTU.EDU.CN
Miaomiao Cheng CHENGMIAOMIAO@BJTU.EDU.CN
Chen Min MINCHEN@BJTU.EDU.CN
Liping Jing LPJING@BJTU.EDU.CN
Beijing Jiaotong University, Beijing, China

Editors: Wee Sun Lee and Taiji Suzuki

Abstract

As a new occurring unsupervised method, multi-view clustering offers a good way to investigate the hidden structure from multi-view data and attracts extensive attention in the community of machine learning and data mining. One popular approach is to identify a common latent subspace for capturing the multi-view information. However, these methods are still limited due to the unsupervised learning process and suffer from considerable noisy information from different views. To address this issue, we present a novel multi-view subspace clustering method, named self-supervised deep multi-view subspace clustering (**S2DMVSC**). It seamlessly integrates spectral clustering and affinity learning into a deep learning framework. **S2DMVSC** has two main merits. One is that the clustering results can be sufficiently exploited to supervise the latent representation learning for each view (via a classification loss) and the common latent subspace learning (via a spectral clustering loss) for multiple views. The other is that the affinity matrix among data objects is automatically computed according to the high-level and cluster-driven representation. Experiments on two scenarios, including original features and multiple hand-crafted features, demonstrate the superiority of the proposed approach over the state-of-the-art baselines.

Keywords: Multi-View Clustering, Subspace Clustering, Deep Learning, Unsupervised Learning

1. Introduction

Nowadays, data with multiple views are becoming more and more popular in many real-world applications. For example, one specific news is described by multilingual forms; an image can be characterized by color, edge, texture, etc. As each view usually contains different and partly independent information, the multi-view learning is beneficial to boost the performance of data analysis such as clustering, classification, retrieval, etc. by fully exploiting the complementary and consistency among different views. In this work, we mainly focus on multi-view clustering, which is a challenging problem for lacking supervised information to guide the learning process.

Multi-view clustering aims to divide the multi-view data into different groups. It has received considerable attention in the area of artificial intelligence and machine learning, because multi-view clustering is superior to single-view clustering by utilizing the complementary information from different views. To make use of the multi-view information, a

surge of common latent subspace methods have been proposed based on different techniques, e.g., canonical correlation analysis (CCA) Chaudhuri et al. (2009); Rupnik and Shawe-Taylor (2010), matrix factorization Gao et al. (2013); Zhang et al. (2014); Zhao et al. (2017) and self-expressive learning Cao et al. (2015); Luo et al. (2018); Cheng et al. (2018). Even though the existing approaches have led to the state-of-the-art multi-view clustering performance, their applicability to real applications is still limited due to the lack of effective supervision. Besides, noise and outlying entries are generally mixed in original data, which adversely degenerate the clustering performance.

To address the above issues, in this paper, we attempt to integrate deep multi-view subspace representation learning and spectral clustering into one unified optimization framework (**S2DMVSC**) as shown in Figure 1. More specifically, to learn better representation for each view and the common latent subspace, **S2DMVSC** supervises such process via two losses, i.e., a spectral clustering loss and a classification loss. To denoise the imperfect correlations among data points, **S2DMVSC** constructs the affinity matrix according to the high-level and cluster-driven representation. These two parts are alternately refined in the learning procedure so that an improved common latent representation can be generated and consequently produces a better data segmentation.

The key contributions of our work are summarized as follows:

- **S2DMVSC** has the ability to obtain a better latent representation by designing a self-supervision framework.
- **S2DMVSC** can seamlessly capture the relationships among multiple views by designing a self-expressive layer between encoder and decoder.
- **S2DMVSC** integrates affinity learning and spectral clustering into a unified framework, which can denoise the imperfect similarity metrics and further improve the final clustering performance.
- Extensive experiments on both the original features and multiple hand-crafted features scenarios demonstrate its superiority by comparing it with the state-of-the-art multi-view clustering methods.

The rest of this paper is organized as follows. A brief survey of multi-view clustering methods is given in section 2. Section 3 shows the proposed model. Section 4 presents experiment results and discussions. Finally, section 5 concludes this paper.

2. Related work

Multi-view clustering is an important and hot research topic in multi-view data analysis. To date, various methods have been proposed by assuming the data points are drawn from multiple low-dimensional subspaces. Based on different techniques, existing subspace methods can be roughly divided into three groups. The first kind of method takes advantage of canonical correlation analysis (CCA) to exploit the consistency of different views by maximizing their correlation. The second kind of method applies non-negative matrix factorization to seek a common latent factor among all views. The third kind of method

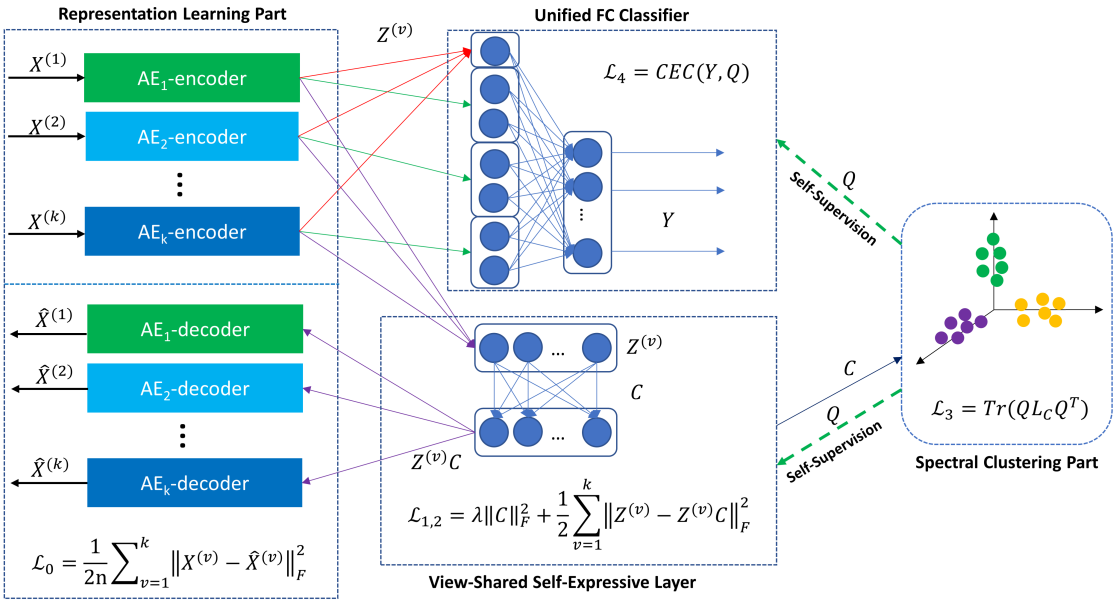

Figure 1: Architecture of the proposed Self-Supervised Deep Multi-View Subspace Clustering (S2DMVSC). It consists of mainly four parts: a) representation learning part, which consists of multiple auto-encoders and used to implement non-linear mapping and learn powerful representations by minimizing reconstruction errors; b) view-shared self-expressive layer, which is used to learn the self-representation coefficient matrix and also takes the self-supervision information from the result of spectral clustering to refine the self-expression coefficient matrix; c) unified FC classifier, which builds a self-supervision path back to the encoder part of deep auto-encoders; d) spectral clustering part, which provides self-supervision information to guide the learning of data representation and view-shared self-expression coefficient.

takes advantage of self-expressive learning to learn the relationship between data points, then the affinity matrix can be constructed to perform spectral clustering.

Chaudhuri et al. (2009) took advantage of the canonical correlation analysis (CCA) to ensure that the new low-dimensional representations of different views are maximally correlated. Rupnik and Shawe-Taylor (2010) extended CCA to more than two views scenario. In the real world, practical data do not necessarily conform with the linear subspace model. Bach and Jordan (2003) took advantage of kernel trick to tackle this drawback, however, how to choose proper kernel is another problem. The other straightforward remedy is based on the matrix factorization technique. Gao et al. (2013) exploited non-negative matrix factorization (NMF) to learn the view-specific new representation of each view and then made a consensus low-dimensional representation. Zhao et al. (2017) took advantage of Semi-NMF to construct multi-layer mapping then get a common latent factor. However, these methods only focus on the low-level relationship among the original features in different views. Another method is to exploit self-expressive learning so that the rela-

tionship between data points can be obtained. Gao et al. (2015) gained a common cluster indicator matrix after learning view-specific self-expression coefficients. Cao et al. (2015) exploited the Hilbert Schmidt Independence Criterion to constrain the diversity among different views. Luo et al. (2018) divided the self-expression coefficient into view-shared and view-specific parts to exploit the complementary and consistent information of multi-view data simultaneously. However, these methods still suffering from the linear mapping which cannot capture complex structures of real-world data.

Andrew et al. (2013) exploited simple fully-connected neural networks to seek non-linear transformation, then utilized CCA to maximize the correlation among different views. Wang et al. (2015) took advantage of deep auto-encoders to learn better latent representations for its reconstruction constraint. Tang et al. (2018) utilized two auto-encoders that introduce the self-expressive layer after CCA at the junction of them. However, it limited to two views. Recently, Abavisani and Patel (2018) exploited multiple deep auto-encoders with one common self-expressive layer to get the unified self-expressive coefficient.

Even though the aforementioned methods obtain promising results, they didn't take the noise ubiquitous in real-world datasets into account. The noise and outlying entries in the real-world datasets will impair the clustering performance. Thus, in this paper, we aim to simultaneously capture complex structures and reduce the effect of noise on clustering performance by introducing supervised information.

3. Self-Supervised Deep Multi-View Subspace Clustering

Self-expressive-based methods Ji et al. (2014, 2017); Zhou et al. (2018) has been proved to be working well for subspace clustering. Given the multi-view dataset $\{X^{(v)} \in \mathbb{R}^{d^{(v)} \times n}\}_{v=1}^{k}$ which has n data points and is described by k views, each data point in v-th view is represented by $d^{(v)}$-dimensional feature vector, and all data points belong to c clusters. If we perform simple self-expressive-based method on each view independently, for v-th view, we have

$$\min_{C^{(v)}} \lambda \|C^{(v)}\|_F^2 + \frac{1}{2}\|X^{(v)} - X^{(v)}C^{(v)}\|_F^2 \quad \text{s.t.} \;\; diag(C^{(v)}) = \mathbf{0}, \tag{1}$$

where $C^{(v)}$ is the v-th view subspace representation, $\|C^{(v)}\|_F^2$ is a regularization term, $diag(C^{(v)}) = \mathbf{0}$ is the constraint to avoid the trivial solution of $C^{(v)} = I$, and $\lambda > 0$ is the trade-off parameter. Once subspace representation $C^{(v)}$ for each view is obtained, how to combine them is challenging. To exploit the consistency among different views, we directly learn an view-shared subspace representation C. Therefore, our model can be reformulated as:

$$\min_{C} \lambda \|C\|_F^2 + \frac{1}{2}\sum_{v=1}^{k} \|X^{(v)} - X^{(v)}C\|_F^2 \quad \text{s.t.} \;\; diag(C) = \mathbf{0}, \tag{2}$$

where C is the view-shared subspace representation, $\|C\|_F^2$ is a regularization term, $diag(C) = \mathbf{0}$ is the constraint to avoid the trivial solution of $C = I$, and $\lambda > 0$ is the trade-off parameter.

However, the above learned linear subspace is not powerful enough to be applied to complex real-world applications. To address this issue, we introduce the deep auto-encoder

structure to exploit the non-linear mapping. The auto-encoder structure consists of two mirror parts: encoder and decoder. For v-th view, given data $X^{(v)}$, feed it into encoder, we can get the latent representation $Z^{(v)}$. To ensure the learned representation $Z^{(v)}$ has meaningful information of original data, $Z^{(v)}$ is fed into the mirror decoder to reconstruct data $\widehat{X}^{(v)}$, then $\widehat{X}^{(v)}$ and $X^{(v)}$ are constrained to be consistent. The key idea is that the learned latent representation $Z^{(v)}$ is a good representation if it can reconstruct original data via the decoder. Therefore, we add the reconstruction error to Eq. 2, it can be rewritten as:

$$\min_{Z^{(v)},C} \frac{1}{2n}\sum_{v=1}^{k}\|X^{(v)}-\widehat{X}^{(v)}\|_F^2+\gamma_1\|C\|_F^2+\frac{\gamma_2}{2}\sum_{v=1}^{k}\|Z^{(v)}-Z^{(v)}C\|_F^2, \quad \text{s.t. } diag(C)=\mathbf{0}, \tag{3}$$

where $Z^{(v)}$ is the v-th view latent representation corresponding to the output of deep encoder, $\widehat{X}^{(v)}$ is the v-th view reconstructed representation of $X^{(v)}$, the first term is reconstruction error, γ_1 and γ_2 are the trade-off parameters among three terms. After obtaining unified subspace representation C, affinity matrix can be constructed as $S=(|C|+|C^T|)/2$ and the clustering result can be obtained by performing spectral clustering on S.

Another important issue for multi-view clustering is that multi-view data contains complex noises. To address this issue, we want to design a constraint between the clustering task and representation learning in a self-supervised way. Fortunately, by observing the fact that the learned output of deep encoders $Z^{(v)}$ (v=1,2,...,k) should contain enough information to produce pseudo label of data, and the output of spectral clustering also can be used to form binary label of data by performing k-means clustering, we design a novel constraint which can guide the learning of latent representations $Z^{(v)}$ (v=1,2,...,k) and the view-shared subspace representation matrix C. More specifically, we design a unified two-layer fully-connected classifier, which takes the output of deep encoders as input. To exploit both complementary and consistent information of different views, we further divide the first fully-connected layer into k view-specific parts and one view-shared part. We denote the output of the unified classifier as $Y\in\mathbb{R}^{n\times c}$. To obtain the output of spectral clustering which takes the affinity matrix S as input, we add the spectral clustering loss to Eq. 3. The objective function of spectral clustering is defined as:

$$\min_{Q} Tr(Q^T L_C Q) \quad \text{s.t. } Q^TQ=I, \tag{4}$$

where $Q\in\mathbb{R}^{n\times c}$ is the cluster indicator matrix, $L_C=D-S$ is a graph Laplacian matrix, D is a diagonal matrix whose diagonal elements are defined as $d_{jj}=\sum_i s_{ij}$. After obtaining the output of spectral clustering, we can get the binary clustering label by performing the k-means algorithm on the row of Q. After Y and Q are obtained, we construct the classification loss by combining cross-entropy loss and center loss (CEC):

$$CEC(Y,Q)=\frac{1}{n}\sum_{j=1}^{n}(\ln(1+e^{-q_j^T\tilde{y}_j})+\tau\|y_j-\mu_{\pi(y_j)}\|_2^2), \tag{5}$$

where y_j is the j-th row of Y, q_j is the j-th row of Q, $\tilde{y}_j$ is the softmax output of y_j, $\mu_{\pi(y_j)}$ is the cluster center which corresponds to y_j, $\pi(y_j)$ is the index of y_j corresponds

to output of spectral clustering Q, and $0 \leq \tau \leq 1$ is the trade-off parameter between two terms. The first cross-entropy loss term to ensure the pseudo label Y and clustering label Q be consistent, the second center loss term to minimize the intra-cluster variations.

Finally, we add Eq. 4 and Eq. 5 to Eq. 3, our model can be rewritten as:

$$\begin{aligned} \min_{Z^{(v)},C,Q} \quad & \frac{1}{2n}\sum_{v=1}^{k}\|X^{(v)}-\widehat{X}^{(v)}\|_F^2+\gamma_1\|C\|_F^2+\frac{\gamma_2}{2}\sum_{v=1}^{k}\|Z^{(v)}-Z^{(v)}C\|_F^2 \\ & +\gamma_3 Tr(Q^T L_C Q)+\frac{\gamma_4}{n}\sum_{j=1}^{n}(\ln(1+e^{-q_j^T\tilde{y}_j})+\tau\|y_j-\mu_{\pi(y_j)}\|_2^2), \\ \text{s.t.} \quad & diag(C)=\mathbf{0}, Q^TQ=I. \end{aligned} \tag{6}$$

where γ_3 and γ_4 are the trade-off parameters for balance spectral clustering term and CEC loss.

To investigate the supervision information generated by the output of spectral clustering, we observe this following formulation which based on the fact that $S=(|C|+|C^T|)/2$,

$$\min Tr(Q^T L_C Q)=\frac{1}{2}\sum_{i,j}s_{ij}\|q_i-q_j\|_2^2=\sum_{i,j}|c_{ij}|\frac{\|q_i-q_j\|_2^2}{2}, \tag{7}$$

once the Q is fixed, c_{ij} will be enforced to non-zero value only if q_i and q_j are the same. We can see the output of spectral clustering can supervise the self-expressive coefficient learning. Meanwhile, we observe that it can supervise the representation learning through the unified FC classifier via CEC loss. Moreover, the output of representation learning $Z^{(v)}(v=1,2,...,k)$ can directly affect the self-expressive learning, whose output (i.e., the unified self-expressive coefficient C) is used to construct the input of spectral clustering. Thus, there is a self-supervision mechanism among representation learning, self-expressive learning, and spectral clustering. The effect of noise on clustering performance can be effectively reduced via this self-supervision mechanism.

3.1. Training Strategy

In the training procedure, we first pre-train all deep auto-encoders only using reconstruction error by setting the trade-off parameters γ_1, γ_2, γ_3 and γ_4 to be zero. Once we obtained pre-trained deep auto-encoders, we can initialize the whole network and train the end-to-end trainable complete model with self-supervision using overall loss in Eq. 6. Because the binary label obtained from the cluster membership Q, i.e., the output of spectral clustering, is based on an unknown permutation, the label information obtained from two successive clusterings might not be consistent. To address this issue, Hungarian algorithm Munkres (1957) is exploited to align the current clustering result with the last one. Besides, we update the output of spectral clustering Q every T_0 iterations and stop training by setting a maximum iteration T_{max}. Furthermore, we use Adam algorithm Kingma and Ba (2015) to minimize the loss function for pre-training and fine-tuning procedures with a learning rate of 1.0×10^{-3} in all our experiments. For clarity, we present the detailed training procedure of proposed S2DMVSC in Algorithm 1.

Algorithm 1 Training procedure of proposed S2DMVSC

Input: Input data $\{X^{(v)} \in \mathbb{R}^{d^{(v)} \times n}\}_{v=1}^{k}$, trade-off parameters γ_1, γ_2, γ_3, γ_4, τ, update iteration T_0 and maximum iteration T_{max}.

Output: trained S2DMVSC and Q.

Pre-train the deep auto-encoders by setting $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 0$;
Initialize other parts of model network using random initialization;
Set $t = 0$;
while $t \leq T_{max}$ **do**
  if $t\%T_0 == 0$ **then**
    Update Q by solving Eq. 4;
  end
  Update whole network via Adam algorithm;
  $t \leftarrow t + 1$;
end

3.2. Computational Complexity Analysis

In this section, we analyze the computational complexity of **S2DMVSC**. Our deep subspace model is composed of two stages, i.e., pre-training and fine-tuning, so we analyze them separately. To simplify the analysis, we assume the settings of all deep auto-encoders (e.g., SAE Vincent et al. (2010)) are the same, and the dimensions of all the layers in each auto-encoder are the same, denoting p. m is the number of encoder layers. The original feature dimensions of each view are the same, denoting d. k is the number of views. n is the number of samples.

In pre-training stage, the complexity of training k m-layer auto-encoders is of order $\mathcal{O}(knT_p(dp+(m-1)p^2))$, where T_p is the maximum number of training iterations. Normally, $p < d$, so the time cost of pre-training stage is $\mathcal{T}_{pre} = \mathcal{O}(knT_{pre}mdp)$. In fine-tuning stage, the training of auto-encoders and spectral clustering are the time consuming parts. Similar to pre-training stage, the time cost of training k m-layer auto-encoders is of order $\mathcal{O}(knT_{max}mdp)$. The time cost of the eigenvalue decomposition in spectral clustering is of order $\mathcal{O}((\lfloor T_{max}/T_0 \rfloor + 1)n^3)$, where T_{max} is the maximum number of Iterations. so the time cost of fine-tuning stage is $\mathcal{T}_{fine} = \mathcal{O}(knT_{max}mdp + (\lfloor T_{max}/T_0 \rfloor + 1)n^3)$. To sum up, the overall computational cost is $\mathcal{T}_{total} = \mathcal{T}_{pre} + \mathcal{T}_{fine}$.

4. Experiments and Analysis

We implement our method in Python with MXNet-1.4.1 Chen et al. (2015), and conduct a series of experiments to validate the performance of the proposed **S2DMVSC** model by comparing with several the-state-of-the-art baseline methods. To validate the effectiveness of our model, we set two kinds of scenarios: original features and multiple hand-crafted features. For original features scenario, to validate that multiple features can help the clustering task, we use raw image pixels as the input and use three different types of auto-encoders to learn three different types of latent representations, which can be seen as three views even though the input is raw image pixels. For hand-crafted features scenario, we use the multiple hand-crafted features as the input to validate the effectiveness of the proposed **S2DMVSC** model when dealing with multiple heterogeneous views.

4.1. Datasets

For the two different experiment scenarios, different datasets are adopted to evaluate the clustering performance. For original features scenario, we adopt five widely-used image datasets: Extended Yale B Georghiades et al. (2001), ORL Samaria and Harter (1994), UMIST Woodall et al. (2007), COIL20 S. A. Nene and Murase (1996a) and COIL100 S. A. Nene and Murase (1996b). For hand-crafted features scenario, four benchmark datasets, including ORL, COIL20, UCIHD Asuncion and Newman (2007) and Yale Belhumeur et al. (1996), are used to validate the performance. The statistics of two types datasets are summarized in Table 1 and Table 2, respectively. Following is the detail information of these datasets.

Table 1: Statistics of benchmark datasets for original features scenario.

Dataset	Type	Samples	Image size	Classes
Extended Yale B	Face	2432	48×42	38
ORL	Face	400	32×32	40
UMIST	Face	480	32×32	20
COIL20	Generic Object	1440	32×32	20
COIL100	Generic Object	2880	32×32	40

Table 2: Statistics of benchmark datasets for hand-crafted features scenario.

Dataset	Type	Samples	Classes	View1	View2	View3
ORL	Face	400	40	Intensity(4096)	LBP(3304)	Gabor(6750)
COIL20	Generic Object	1440	20	Intensity(1024)	LBP(3304)	Gabor(6750)
UCIHD	Digit	2000	10	FAC(216)	FOU(76)	PIX(240)
Yale	Face	165	15	Intensity(4096)	LBP(3304)	Gabor(6750)

- *Extended Yale B*: it contains 2432 face images of 38 subjects, the images of each subject were taken under different illumination conditions. Following Ji et al. (2017), each image is down-sampled to 48 × 42.
- *ORL*: it composed of 400 face images taken from 40 individuals. For original features scenario, following Ji et al. (2017), each image is down-sampled to 32 × 32. For hand-crafted features scenario, each image is represented by three kinds of features (4096 Intensity, 3304 LBP and 6750 Gabor).
- *UMIST*: it contains 480 images of 20 persons, each person with only 24 images is taken under very different poses. Each image of the dataset is down-sampled to 32 × 32.
- *COIL20*: it contains 1440 gray-scale images of 20 objects. For original features scenario, each image is down-sampled to 32 × 32. For multiple hand-crafted features scenario, three types of features are extracted including Intensity, LBP and Gabor. Their feature size is (1024, 3304 and 6750).
- *COIL100*: it is composed of 7200 images taken from 100 objects. Due to the computational memory limit, following Zhou et al. (2018), we select the first 40 classes of 2880 data points in COIL100 for our experiments. Each image is down-sampled to 32 × 32.

- *UCIHD*: it contains 2000 images of 10 categories, and each image is represented by three kinds of features: 216 Profile-correlation, 76 Fourier-coefficient and 240 Intensity-averaged.
- *Yale*: it is composed of 165 images taken from 15 individuals. Three types of features are extracted including Intensity, LBP and Gabor. Their feature size is (4096, 3304 and 6750) respectively.

4.2. Methodology

To prove the effectiveness of proposed **S2DMVSC**, we compare it with several state-of-the-art baselines under two different experiment scenarios. In original features scenario, i.e., raw image pixels as the input, six baselines, which use original features as input data, are adopted for comparisons. Among them, SSC Elhamifar and Vidal (2013) and EDSC Ji et al. (2014) to learn the self-expression coefficient matrix by using sparsity prior, LRR Liu et al. (2010) and LRSC Vidal and Favaro (2014) to pursue better self-expression coefficient by constraining it as low-rank as possible, all these methods are using shallow features with linear mapping. Meanwhile, three deep subspace clustering networks are compared: DSC-Nets (including DSC-Net-l_1 and DSC-Net-l_2) Ji et al. (2017) and DASC Zhou et al. (2018). DSC-Nets introduce the self-expression method to deep auto-encoders, utilize deep auto-encoder to exploit non-linear mapping and learn the self-expression coefficient to perform spectral clustering. DASC improves DSC-Nets by introducing Generative Adversarial Nets Goodfellow et al. (2014) to refine the feature representations learned from auto-encoder.

In multiple hand-crafted features scenario, i.e., multiple hand-crafted features as the input of model, six state-of-the-art baselines, including DiMSC Cao et al. (2015), CSMSC Luo et al. (2018), SwMC Nie et al. (2017), MCGC Zhan et al. (2019), DMF-MVC Zhao et al. (2017) and DMSC Abavisani and Patel (2018). Among them, DiMSC first learns the view-specific self-expression coefficient and then introduces the Hilbert Schmidt Independence Criterion Gretton et al. (2005) to constrain the diversity among different views. CSMSC divides the self-representation coefficient matrix into view-shared and view-specific parts so as to exploit the complementary and consistent information of multi-view data simultaneously. SwMC and MCGC aim to fusion multiple view-specific affinity matrix so as to exploit the complementary and consistency information among different views. Meanwhile, DMF-MVC constructs multi-layer semi-non-negative matrix factorization to learn a deep consensus latent representation for all views. DMSC extends DSC-Nets to multi-view applications, and uses multiple auto-encoders with a common self-expressive layer to exploit the consistency among multiple views. Because the original DMSC uses multiple deep auto-encoders to solve multimodal image clustering, which is not the same as our experiment setting, we keep its network settings the same as ours for fair comparison. Besides, recently published DMVSSC Tang et al. (2018) adopts two convolutional auto-encoders to extract the feature, then utilizes CCA to fusion the feature representations, finally performs affinity learning by self-expression layer. However, it is limited to two views cases, so we do not take it as baselines.

In our experiments, the optimal parameter settings keep the same as previous suggestions or determined by the experiments. For all baseline methods, we use the source codes released by the corresponding paper authors, and tune parameters to get the best results.

Note that we directly cite the results of some baselines, including SSC, EDSC, LRR, LRSC, DSC-Net-l_1, DSC-Net-l_2 and DASC, from DASC under the original features scenario due to the same experiment settings. In all our experiments, we choose ReLU Krizhevsky et al. (2012) as the non-linear activation function, we set the learning rate to 1.0×10^{-3} and we set the trade-off parameter $\tau = 1$ between cross-entropy loss and center loss. For learning the self-expression coefficient matrix, the whole dataset is fed into the model as one batch. In addition, for the unified FC layers, the first hidden layer consists of three view-specific parts which have $n/3$ neurons each, and one view-shared part which has $n/50$ neurons, $51n/50$ neurons in total, and the second FC layer has c neurons.

To measure the clustering performance, we use the following popular clustering metrics: clustering accuracy (ACC), normalized mutual information (NMI) Vinh et al. (2010) and purity (PUR) Manning et al. (2008). The values of all the three metrics are in the range $[0, 1]$, and the higher the better.

4.3. Performance Evaluation

In this section, we evaluate the proposed **S2DMVSC** from four aspects. Firstly, we present the effect of parameters on our S2DMVSC. We show the effectiveness of S2DMVSC when dealing with the original features scenario by comparing it with several baselines in the second part. The third one is to present the superiority of S2DMVSC on multiple hand-crafted features scenario by comparing with six baselines. Finally, we check the convergence of our proposed S2DMVSC.

4.3.1. Effect of parameters

In our model, the architecture of deep auto-encoder networks followed by the previous suggestions or determined empirically. There are five trade-off parameters, including γ_1, γ_2, γ_3, γ_4 and τ, may affect the performance of S2DMVSC. τ is fixed to 1 in experiments. For the common parameters γ_1 and γ_2 on all datasets (except for UCIHD and Yale due to absence), we keep the same as Zhou et al. (2018) and Ji et al. (2017). For other experiments, we tune γ_1, γ_2, γ_3 and γ_4 in $\{0.01, 0.05, 0.1, 0.5, 1, 10, 30, 50, 70, 100\}$. The detailed parameter settings are reported on Table 5 and Table 6.

4.3.2. Evaluation under Original Features Scenario

For original features scenario, three different deep auto-encoders, including SAE Vincent et al. (2010), CAE Guo et al. (2017), and CVAE Kulkarni et al. (2015), are adopted to learn latent representations as three views, and the kernel stride of deep convolutional auto-encoders (i.e., CAE and CVAE) is fixed to 2. The detailed architecture information of CAE, CVAE is summarized in Table 7. Note that the main architectures of CVAE are the same as CAE and the latent dimension of CVAE is fixed to 256. The architecture of SAE is fixed to three-layer with {500,500,2000} dimensions for all datasets. For clarity, the detailed setting information of parameters, including trade-off parameters γ_1, γ_2, γ_3, γ_4 in Eq. 6, update iteration T_0 and maximum iteration T_{max}, is summarized in Table 5.

In this subsection, we will evaluate the effectiveness of our proposed method when dealing with original features scenario, where raw pixels of image datasets are used as the input of model, by comparing with six baselines on five benchmark datasets. For all

Table 3: Clustering results of original features scenario.

Datasets	Metrics	SSC	EDSC	LRR	LRSC	DSC-Net-l_1	DSC-Net-l_2	DASC	S2DMVSC
Extended Yale B	ACC	0.7354	0.8814	0.8499	0.7931	0.9681	0.9733	0.9856	**0.9906**
	NMI	0.7796	0.8835	0.8636	0.8264	0.9687	0.9703	0.9801	**0.9912**
	PUR	0.7467	0.8800	0.8623	0.8013	0.9711	0.9731	0.9857	**0.9908**
ORL	ACC	0.7425	0.7038	0.8100	0.7200	0.8550	0.8600	0.8825	**0.8960**
	NMI	0.8459	0.7799	0.8603	0.8156	0.9023	0.9034	0.9315	**0.9425**
	PUR	0.7875	0.7138	0.8225	0.7542	0.8585	0.8625	0.8925	**0.9065**
UMIST	ACC	0.6904	0.6937	0.6979	0.6729	0.7242	0.7312	0.7688	**0.7900**
	NMI	0.7489	0.7522	0.7630	0.7498	0.7556	0.7662	0.8042	**0.8352**
	PUR	0.6554	0.6683	0.6670	0.6562	0.7204	0.7276	0.7688	**0.7908**
COIL20	ACC	0.8631	0.8371	0.8118	0.7416	0.9314	0.9368	0.9639	**0.9796**
	NMI	0.8892	0.8828	0.8747	0.8452	0.9353	0.9408	0.9686	**0.9806**
	PUR	0.8747	0.8585	0.8361	0.7937	0.9306	0.9397	0.9632	**0.9788**
COIL100	ACC	0.7191	0.6870	0.6493	0.6327	0.8003	0.8075	0.8354	**0.8660**
	NMI	0.8212	0.8139	0.7828	0.7737	0.8852	0.8941	0.9196	**0.9456**
	PUR	0.7716	0.7469	0.7109	0.6981	0.8646	0.8740	0.8972	**0.9295**

Table 4: Clustering results of multiple hand-crafted features scenario.

Datasets	Metrics	DiMSC	SwMC	CSMSC	MCGC	DMF-MVC	DMSC	S2DMVSC
ORL	ACC	0.8380	0.7075	0.8680	0.7975	0.7780	0.8732	**0.9004**
	NMI	0.9400	0.8237	0.9420	0.9006	0.8768	0.9402	**0.9500**
	PUR	0.8875	0.7675	0.8900	0.8350	0.8080	0.8940	**0.9020**
COIL20	ACC	0.8007	0.8639	0.7729	0.9951	0.8105	0.9512	**0.9965**
	NMI	0.8505	0.9429	0.8548	0.9945	0.9160	0.9503	**0.9963**
	PUR	0.8021	0.8986	0.7979	0.9951	0.8590	0.9515	**0.9967**
UCIHD	ACC	0.7965	0.8555	0.8880	0.9705	0.8112	0.9542	**0.9715**
	NMI	0.7471	0.8910	0.8134	0.9301	0.8311	0.9020	**0.9374**
	PUR	0.7965	0.8800	0.8880	0.9705	0.8543	0.9542	**0.9715**
Yale	ACC	0.7090	0.6545	0.7520	0.7152	0.7467	0.7273	**0.7750**
	NMI	0.7270	0.6872	0.7840	0.6734	0.7820	0.7678	**0.7922**
	PUR	0.6970	0.6545	0.7621	0.7152	0.7558	0.7273	**0.7785**

Table 5: Parameter settings under original features scenario

Layers	γ_1	γ_2	γ_3	γ_4	T_0	T_{max}
Extended Yale B	1	1	30	70	5	1000
ORL	0.1	0.01	10	1	5	1240
UMIST	1	0.1	10	30	5	560
COIL20	1	15	10	10	4	320
COIL100	1	15	10	10	8	720

Table 6: Parameter settings under multiple hand-crafted features scenario.

Layers	γ_1	γ_2	γ_3	γ_4	T_0	T_{max}
ORL	0.1	0.01	10	1	5	360
COIL20	1	15	10	10	5	40
UCIHD	10	30	10	30	4	205
Yale	1	30	0.05	30	5	30

experiments, we run ten times and get the average results. The best result is marked in bold and the second is underlined for each metric.

Table 3 reports the comparison experiment results. The improvements that **S2DMVSC** achieves relative to seven baselines on five datasets in terms of NMI are presented in Figure 2. The relative improvement in terms of NMI between two methods is calculated by Relative Improvement $= (NMI_1 - NMI_2)/NMI_2$, where NMI_1 refers to the NMI result obtained by our proposed **S2DMVSC** and NMI_2 indicates the NMI obtained by the corresponding baselines. From the above comparison experiment results, we can obtain the following points. 1) On each benchmark dataset, the improvement of deep structure-based methods (i.e., DSC-Net-l_1, DSC-Net-l_2, DASC and **S2DMVSC**) over the traditional mod-

Table 7: Network settings of CAE and CVAE for original features scenario.

Datasets	Layers	encoder-1	encoder-2	encoder-3	decoder-1	decoder-2	decoder-3
Extended Yale B	channels	10	20	30	30	20	10
	kernel size	5×5	3×3	3×3	3×3	3×3	5×5
ORL	channels	5	3	3	3	3	5
	kernel size	5×5	3×3	3×3	3×3	3×3	5×5
UMIST	channels	15	10	5	5	10	15
	kernel size	5×5	3×3	3×3	3×3	3×3	5×5
COIL20	channels	15	-	-	15	-	-
	kernel size	3×3	-	-	3×3	-	-
COIL100	channels	20	-	-	20	-	-
	kernel size	3×3	-	-	3×3	-	-

els (i.e., SSC, EDSC, LRR and LRSC) shows that deep structure network can effectively exploit the non-linear mapping of complex real-world datasets and extract powerful feature representations. 2) As expected, **S2DMVSC** consistently outperforms all the deep structure baselines in terms of ACC, NMI, and PUR on each benchmark dataset. The main reason is that the multiple different deep auto-encoders designed in our model can learn diverse features from different views, the complementary information can significantly improve the clustering performance.

To prove the advantage of our method when dealing with multiple hand-crafted features scenario, we further evaluate the clustering performance of **S2DMVSC** on multiple hand-crafted features datasets next subsection.

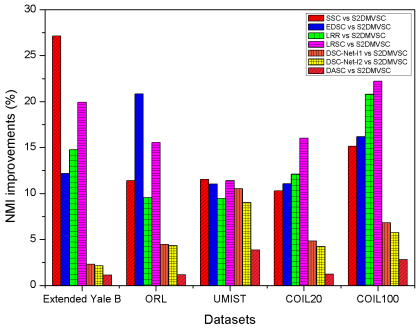

Figure 2: The improvement percentage between S2DMVSC and baselines under original features scenario in terms of NMI.

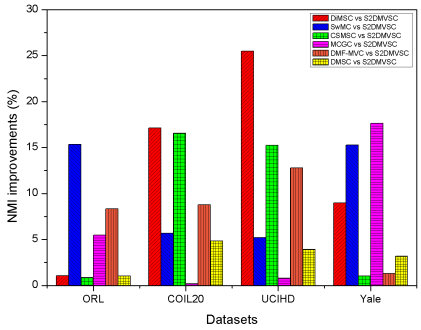

Figure 3: The improvement percentage between S2DMVSC and baselines under multiple hand-crafted features scenario in terms of NMI.

4.3.3. Evaluation under Multiple Hand-crafted Features Scenario

For multiple hand-crafted features scenario, we use the same SAE for all views to exploit non-linear mapping. For ORL, UCIHD and Yale, we use one-layer SAE with {400}, {300} and {400} dimensions respectively, and use three-layer SAE for COIL20 with {500,500,2000} dimensions. Besides, the detailed setting information of the trade-off parameters γ_1, γ_2, γ_3 and γ_4 in overall loss function Eq. 6, update iteration T_0 and maximum iteration T_{max} is summarized in Table 6.

In this subsection, we will evaluate the effectiveness of our proposed method when dealing with multiple hand-crafted features scenario, where data points are described by multiple handcrafted features, by comparing with six baselines on four benchmark datasets. For all experiments, we run ten times and get the average results. The best result is marked in bold and the second is underlined for each metric.

Table 4 reports the comparison experiment results between **S2DMVSC** and six baselines on four benchmark datasets. From the comparison results, we can observe that our method outperforms DMSC, which keeps the same deep auto-encoders and unified self-expressive layer with our model. It is because the self-supervision information produced by the output of spectral clustering can effectively help the training of representation learning and unified self-expression layer, leading to better clustering performances. In most cases, CSMSC outperforms DiMSC, SwMC, and DMF-MVC. It shows the importance of exploiting the complementary and consistency among all views simultaneously. Comparing with MCGC, we can show the strength of deep auto-encoders used in our model to learn effective representations in an unsupervised manner. Besides, we can further observe that our **S2DMVSC** yields the best results comparing with all the baselines in terms of all three metrics on each dataset. The improvements that **S2DMVSC** achieves relative to six baselines on four datasets in terms of NMI are shown in Figure 3. These comparison results show the effectiveness of our method.

4.3.4. Convergence of S2DMVSC

To verify the efficiency of proposed **S2DMVSC**, we record the objective value (Eq. 6) and the clustering performance (NMI) of the fine-tuning stage along with the iterations on UCIHD dataset in Figure 4. We can observe that the objective value decreases quickly with the increasing of iterations in general even though there are some shocks within 30 iterations. This result verifies that the proposed method converges quickly, and yields good clustering results. Similar convergence can be obtained on other datasets.

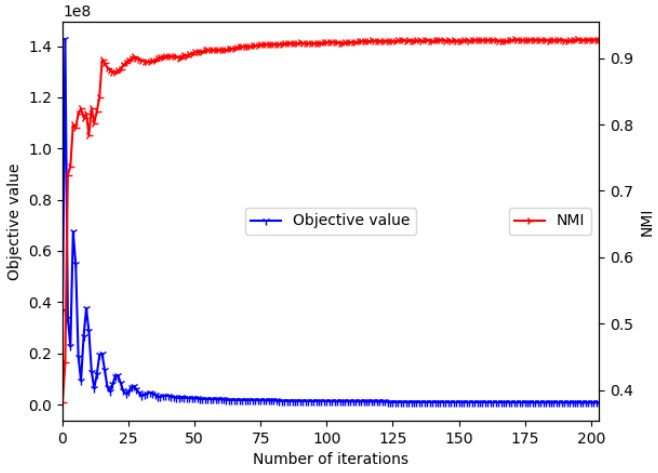

Figure 4: The objective value (blue line) and NMI (red line) with respect to number of iterations on UCIHD dataset.

5. Conclusions

In this paper, we proposed a novel end-to-end trainable multi-view subspace clustering method, named self-supervised deep multi-view subspace clustering (**S2DMVSC**). It seamlessly integrates spectral clustering and affinity learning into a deep learning framework. More specifically, to learn better representation for each view and the common latent subspace, **S2DMVSC** supervises such process via two losses, i.e., a spectral clustering loss and a classification loss. To denoise the imperfect correlations among data points, **S2DMVSC** constructs the affinity matrix according to the high-level and cluster-driven representation. These two parts are alternately refined in the learning procedure so that an improved common latent representation can be generated and consequently produces a better data segmentation. Experiments on two scenarios, including original features and multiple hand-crafted features, demonstrate the superiority of the proposed approach over the state-of-the-art baselines.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61822601, 61773050, and 61632004; the Beijing Natural Science Foundation under Grant Z180006; the Beijing Municipal Science & Technology Commission under Grant Z181100008918012.

References

Mahdi Abavisani and Vishal M. Patel. Deep multimodal subspace clustering networks. *IEEE Journal of Selected Topics in Signal Processing*, 12:1601–1614, 2018.

Galen Andrew, Raman Arora, Jeff A. Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *ICML*, 2013.

Arthur U. Asuncion and Dava J. Newman. Uci machine learning repository. 2007.

Francis R. Bach and Michael I. Jordan. Kernel independent component analysis. In *ICASSP*, pages 876–879, 2003.

Peter N. Belhumeur, João Pedro Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In *ECCV*, 1996.

Xiaochun Cao, Changqing Zhang, Huazhu Fu, Si Liu, and Hua Zhang. Diversity-induced multi-view subspace clustering. In *CVPR*, pages 586–594, 2015.

Kamalika Chaudhuri, Sham M. Kakade, Karen Livescu, and Karthik Sridharan. Multi-view clustering via canonical correlation analysis. In *ICML*, 2009.

Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *ArXiv*, abs/1512.01274, 2015.

Miaomiao Cheng, Liping Jing, and Michael K. Ng. Tensor-based low-dimensional representation learning for multi-view clustering. *IEEE Transactions on Image Processing*, 28: 2399–2414, 2018.

Ehsan Elhamifar and René Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 2013.

Hongchang Gao, Feiping Nie, Xuelong Li, and Heng Huang. Multi-view subspace clustering. In *ICCV*, pages 4238–4246, 2015.

Jing Gao, Jiawei Han, Jialu Liu, and Chi Wang. Multi-view clustering via joint nonnegative matrix factorization. In *SDM*, 2013.

Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23:643–660, 2001.

Ian J. Goodfellow, Jean Pouget-Abadie, Mansha Parven Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

Arthur Gretton, Olivier Bousquet, Alexander J. Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *ALT*, 2005.

Xifeng Guo, Xinwang Liu, En Zhu, and Jianping Yin. Deep clustering with convolutional autoencoders. In *ICONIP*, 2017.

Pan Ji, Mathieu Salzmann, and Hongdong Li. Efficient dense subspace clustering. *IEEE Winter Conference on Applications of Computer Vision*, pages 461–468, 2014.

Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian D. Reid. Deep subspace clustering networks. In *NIPS*, 2017.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60:84–90, 2012.

Tejas D. Kulkarni, William F. Whitney, Pushmeet Kohli, and Joshua B. Tenenbaum. Deep convolutional inverse graphics network. In *NIPS*, 2015.

Guangcan Liu, Zhouchen Lin, and Yong Yu. Robust subspace segmentation by low-rank representation. In *ICML*, 2010.

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

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. 2008.

James Raymond Munkres. Algorithms for the assignment and transportation problems. 1957.

Feiping Nie, Jing Li, and Xuelong Li. Self-weighted multiview clustering with multiple graphs. In *IJCAI*, 2017.

Jan Rupnik and John Shawe-Taylor. Multi-view canonical correlation analysis. 2010.

S. K. Nayar S. A. Nene and H. Murase. Columbia Object Image Library (COIL-20). Technical report, Feb 1996a.

S. K. Nayar S. A. Nene and H. Murase. Columbia Object Image Library (COIL-100). Technical report, Feb 1996b.

Ferdinand Samaria and Andy Harter. Parameterisation of a stochastic model for human face identification. In *WACV*, 1994.

Xiaoliang Tang, Xuan Tang, Wanli Wang, Li Fang, and Xian Wei. Deep multi-view sparse subspace clustering. In *ICNCC*, 2018.

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

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11:3371–3408, 2010.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11:2837–2854, 2010.

Weiran Wang, Raman Arora, Karen Livescu, and Jeff A. Bilmes. On deep multi-view representation learning. In *ICML*, 2015.

Jeremy Woodall, Marcelino Agúndez, Andrew J. Markwick-Kemper, and Tom J Millar. The umist database for astrochemistry 2006. 2007.

Kun Zhan, Feiping Nie, Jing Wang, and Yi Yang. Multiview consensus graph clustering. *IEEE Transactions on Image Processing*, 28:1261–1270, 2019.

Xianchao Zhang, Long Zhao, Linlin Zong, Xinyue Liu, and Hong Yu. Multi-view clustering via multi-manifold regularized nonnegative matrix factorization. In *ICDM*, pages 1103–1108, 2014.

Handong Zhao, Zhengming Ding, and Yun Fu. Multi-view clustering via deep matrix factorization. In *AAAI*, 2017.

Pan Zhou, Yunqing Hou, and Jiashi Feng. Deep adversarial subspace clustering. In *CVPR*, pages 1596–1604, 2018.