Cascade Subspace Clustering*

Xi Peng,[1] **Jiashi Feng,**[2] **Jiwen Lu,**[3] **Wei-Yun Yau,**[1] **Zhang Yi**[4]
[1]Institute for Infocomm Research, A*STAR, Singapore; [2]National University of Singapore, Singapore
[3]Department of Automation, Tsinghua University, Beijing, China
[4]College of Computer Science, Sichuan University, Chengdu, P. R. China.
pangsaai@gmail.com, elefjia@nus.edu.sg, lujiwen@tsinghua.edu.cn, wyyau@i2r.a-star.edu.sg, zhangyi@scu.edu.cn

Abstract

In this paper, we recast the subspace clustering as a verification problem. Our idea comes from an assumption that the distribution between a given sample $\mathbf{x}$ and cluster centers $\mathbf{\Omega}$ is invariant to different distance metrics on the manifold, where each distribution is defined as a probability map (*i.e.* soft-assignment) between $\mathbf{x}$ and $\mathbf{\Omega}$. To verify this so-called *invariance of distribution*, we propose a deep learning based subspace clustering method which simultaneously learns a compact representation using a neural network and a clustering assignment by minimizing the discrepancy between pair-wise sample-centers distributions. To the best of our knowledge, this is the first work to reformulate clustering as a verification problem. Moreover, the proposed method is also one of the first several cascade clustering models which jointly learn representation and clustering in end-to-end manner. Extensive experimental results show the effectiveness of our algorithm comparing with 11 state-of-the-art clustering approaches on four data sets regarding to four evaluation metrics.

Introduction

Data clustering is a popular unsupervised learning technique to analyze unlabeled data (Jain, Murty, and Flynn 1999), which aims to group a collection of samples into different clusters by simultaneously minimizing inter-cluster similarity and maximizing intra-cluster similarity. Two challenging problems in applying clustering in realistic data are the curse of high-dimensionality and linear inseparability of the inherent clusters – which have attracted numerous researches during the past several decades. These two problems are actually two sides of one coin. Specifically, many real-world data such as images and documents are very high dimensional, which are generally believed to be separated better within non-Euclidean space. In other words, it is difficult to separate these data using Euclidean distance based clustering approaches such as vanilla kmeans.

To cluster high-dimensional data, various methods have been proposed (Ng, Jordan, and Weiss 2001; Zhao and Tang 2009; Yu et al. 2015), among which subspace clustering is quite popular (Vidal 2011). Subspace clustering implicitly seeks a low-dimensional subspace to fit each group of data points and separates these data in the projection space with the following two steps: 1) learning low-dimensional representation for a given data set, and 2) clustering data based on the representation. Through exploiting the low-dimensional subspace structure, both the problem of dimensionality curse and linear inseparability can be effectively alleviated.

During the past several years, most existing subspace clustering methods focus on how to learn a good data representation that is beneficial to discover the inherent clusters (Elhamifar and Vidal 2013; Liu et al. 2013; Feng et al. 2014; Wang, Zhu, and Yuan 2014; Hu et al. 2014; Peng, Yi, and Tang 2015; Xiao et al. 2015; Peng et al. 2016c). Like the standard spectral clustering (SC) (Ng, Jordan, and Weiss 2001), those methods cluster data by: 1) building an affinity graph to describe the relationship among data points. 2) using the graph as a prior to learn low-dimensional data representation, and 3) performing kmeans on the obtained representation to obtain clustering results. In fact, the first two steps can be regarded as conducting manifold learning (Belkin and Niyogi 2003; Wang, Lin, and Yuan 2016) which preserves a similarity graph from input space into a low-dimensional one.

Although those subspace clustering methods have shown encouraging performance, we observe that they suffer from the following limitations. First, most subspace clustering methods learn data representation via shallow models which may not capture the complex latent structure of big data. Second, the methods require to access the whole data set as the dictionary, and thus making difficulty in handling large scale and dynamic data set. To solve these problems, we believe that deep learning could be an effective solution thanks to its outperforming representation learning capacity and fast inference speed. In fact, (Peng et al. 2016b; Yang, Parikh, and Batra 2016; Xie, Girshick, and Farhadi 2016) have very recently proposed to learn representation for clustering using deep neural networks. However, most of them do not work in an end-to-end manner which however is generally believed to be the major factor for the success of deep learning (Bengio, Courville, and Vincent 2013; Lecun, Bengio, and Hinton 2015).

In this paper, we propose a novel end-to-end trainable deep subspace clustering method, termed cascade subspace clustering (CSC). Our basic idea comes from an assump-

*Corresponding Author: Zhang Yi.

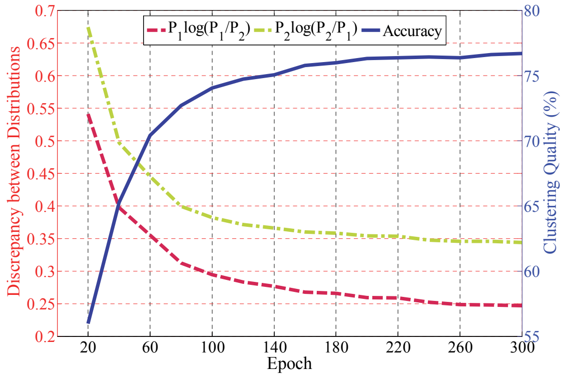


Figure 1: Key observation that motivates our idea. In the plot, the left y-axis indicates the discrepancy between $\mathcal{P}_1$ and $\mathcal{P}_2$ and the right one is the corresponding clustering *Accuracy*, where $\mathcal{P}_1$ and $\mathcal{P}_2$ are two probability maps based on Euclidean and Cosine distance that reflect the distributions between data sets and cluster centers in a 10-dimensional space. Specifically, we project the full mnist data set into a latent space using a 784-500-500-2000-10 encoder, and then calculate $\mathcal{P}_1$, $\mathcal{P}_2$, and their discrepancy based on the KL divergence loss. More details about the experiment can refer to our experiments. With more training epochs for the encoder, ones can see that 1) the clustering accuracy increases from 58% to 77%, and 2) the discrepancy between two distributions $\mathcal{P}_1$ and $\mathcal{P}_2$ monotonically decreases. The plot demonstrates that better representation always leads to smaller discrepancy between $\mathcal{P}_1$ and $\mathcal{P}_1$ and better clustering results.

tion that we called *invariance of distribution*. Figure 1 gives an example to illustrate the idea. In details, for a given data point $\mathbf{x}$, the *conditional distribution* $\mathcal{P}(\mathbf{h}|\mathbf{\Omega})$ *should be invariant* to different distance metrics in the latent space, where $\mathbf{h}$ denotes the representation of $\mathbf{x}$, and $\mathcal{P}(\mathbf{h}|\mathbf{\Omega})$ is the probability map of $\mathbf{h}$ *w.r.t.* cluster centers $\mathbf{\Omega}$. Based on this assumption, *we recast the clustering problem as a variant of verification*. Noticed that, the traditional verification aims to judge whether a given pair of samples belong to the same subject. In contrast, our new formulation models the sample-centers distribution using a collection of "positive" pairs and minimizes the discrepancy between different distributions, where we called "positive" as the pairwise distributions are based on the same data point and cluster centers.

To implement our idea, we propose CSC which first encodes each sample into a latent space and then minimizes the difference between two sample-centers distributions defined by Euclidean and Cosine distance. CSC is a cascade model of which the first step (representation learning) is performed in the forward pathway to map input into a latent space, and the second step (clustering) is performed in the backward pathway to provide a supervision signal for updating the neural network. With this strategy, even no human annotation is provided for the data, the cascade model can still be trained end-to-end and such an end-to-end manner leads to better representation and clustering results. The novelty and contribution of this work could be summarized as follows:

- To the best of our knowledge, *this is the first work to recast the clustering problem as verification*. Although verification has been extensively studied for various supervised learning tasks such as face verification, there is no work trying to bridge clustering and verification. Thus, we believe that this work would provide novel insights and bridge unsupervised clustering and verification.
- The propose CSC is among the first cascade clustering models. Different from existing methods, CSC works in end-to-end rather than plug-in manner. By jointly learning data representation and performing clustering, CSC could give better clustering results and representation. Moreover, unlike most existing subspace clustering algorithms, our algorithm does not require to use the whole data set as the dictionary. This enables our method to perform fast inference and more efficiently handle large scale data sets.

Related Works

Subspace clustering: Benefit from the effectiveness of manifold learning (Belkin and Niyogi 2003), subspace clustering has achieved remarkable developments in various applications such as image segmentation, data clustering, and motion segmentation (Vidal 2011). Most recent subspace clustering methods could be regarded as extensions of spectral clustering (Ng, Jordan, and Weiss 2001). Both all of them learn a compact representation using manifold learning and obtain clustering assignment by performing kmeans on the representation. The main difference among them is the way to learn representation. Since the key of manifold learning based representation learning is similarity graph, most recent subspace clustering methods have focused on how to construct a good graph. These works propose building the graph using the following reconstruction coefficient:

$$\min_{\mathbf{c}_i} \frac{1}{2}\|\mathbf{x}_i - \mathbf{X}\mathbf{c}_i\|_F^2 + \lambda\mathcal{R}(\mathbf{c}_i), \tag{1}$$

where $\mathbf{x}_i$ and $\mathbf{c}_i$ denote the i-th data point of $\mathbf{X}$ and the corresponding self-expression coefficients, respectively. $\|\cdot\|_F$ denotes Frobenius norm, and $\mathcal{R}(\mathbf{c}_i)$ is the adopted prior on $\mathbf{c}_i$. Different works adopt different $\mathcal{R}(\cdot)$ and three of them are most popular, *i.e.* ℓ_1-norm based sparsity (Elhamifar and Vidal 2013; Feng et al. 2014), nuclear-norm based low rankness (Liu et al. 2013; Vidal and Favaro 2014; Xiao et al. 2015), and Frobenius norm based sparsity (Peng et al. 2016a; 2016c).

Unlike those approaches, our method learns representation using neural network instead of manifold learning. This brings several advantages. First, our CSC could handle large scale data set since it obtains data representation without requirements of using the whole data set as dictionary and solving a $n \times n$ (*i.e.* data size) singular value decomposition (SVD) problem. Second, CSC jointly learns representation and performs clustering in end-to-end manner, while these two steps are separately treated by those existing subspace clustering methods. As our method utilizes clustering results as a supervisor, it could learn a better representation. Third,

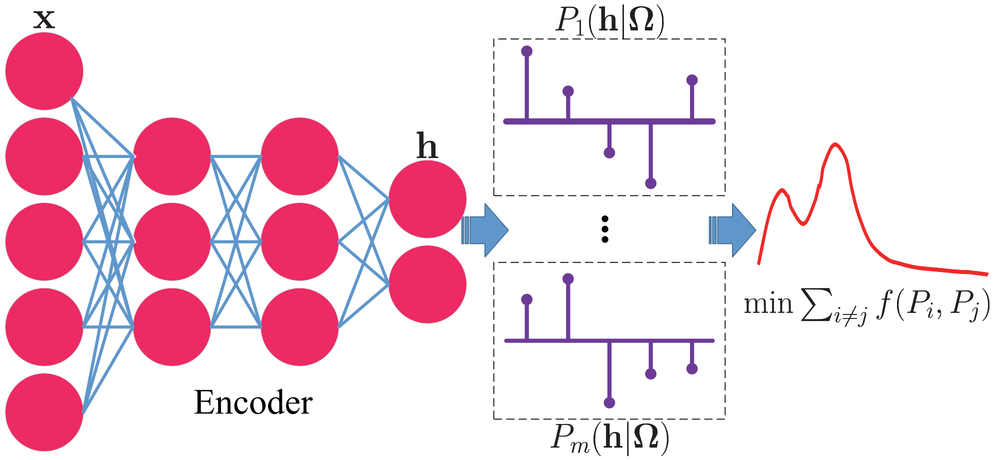

$\mathbf{x}$
Encoder
$\mathbf{h}$
$P_1(\mathbf{h}|\mathbf{\Omega})$
$P_m(\mathbf{h}|\mathbf{\Omega})$
$\min \sum_{i \neq j} f(P_i, P_j)$

Figure 2: The architecture of our CSC. For a given data point $\mathbf{x}$, CSC jointly learns representation and performs clustering in two successive steps, one is an encoder which maps $\mathbf{x}$ into a latent space to get $\mathbf{h}$, and the other is a clustering module which minimizes the discrepancy among different distributions of $\mathbf{h}$ *w.r.t.* cluster centers $\mathbf{\Omega}$. In this work, the encoder is initialized in self-supervised manner (*i.e.* autoencoder) and $\mathbf{\Omega}$ is initialized by kmeans. $\mathcal{P}_1(\mathbf{h}|\mathbf{\Omega}), \mathcal{P}_2(\mathbf{h}|\mathbf{\Omega}), \cdots, \mathcal{P}_m(\mathbf{h}|\mathbf{\Omega})$ are distributions of $\mathbf{h}$ *w.r.t.* $\mathbf{\Omega}$ regarding to m metrics, and the loss $f(\cdot)$ describes the discrepancy between two distributions.

CSC is a deep model which could be more effective to capture the latent structure of complex real-world data set.

Deep Learning: As the most effective representation learning technique, deep learning has been extensively studied for various applications, especially, in the scenario of supervised learning (Krizhevsky, Sutskever, and Hinton 2012; Hu, Lu, and Tan 2014). In contrast, only a few of works have devoted to unsupervised scenario which is one of major challenges faced by deep learning (Bengio, Courville, and Vincent 2013; Lecun, Bengio, and Hinton 2015). Clustering as one of the most important unsupervised learning tasks, fewer works investigate how to make it benefiting from deep learning (Peng et al. 2016b; Yang, Parikh, and Batra 2016; Xie, Girshick, and Farhadi 2016).

Unlike those deep clustering approaches, our method is the first work to recast the clustering as a verification problem. Moreover, some of these approaches obtain results in off-the-shelf manner, whereas our deep model is a clustering-oriented cascade model. It is generally believable that the task specific on-the-shelf deep learning is more promising and attractive (Bengio, Courville, and Vincent 2013; Lecun, Bengio, and Hinton 2015).

Cascade Subspace Clustering

In this section, we first elaborate on the details of the proposed CSC model and then give the implementation details of the algorithm.

The Model of CSC

For a given data set $\mathbf{X} \in \mathbb{R}^{m \times n}$, we aim to assign each data point $\mathbf{x}_i \in \mathbf{X}$ into one of k clusters of which each is represented by a centroid $\omega_j \in \mathbf{\Omega}$, where m denotes the input dimension and n is the data size. To this end, CSC obtains results with two joint modules. One is learning representation with an encoder and the other is clustering data by minimizing the discrepancy among different distributions. Figure 2 gives an illustration to the proposed CSC.

To learn a good representation $\mathbf{H} \in \mathbb{R}^{d \times n}$, CSC progressively maps $\mathbf{X}$ into a low-dimensional space via a series of nonlinear transformations. Here, d denotes dimension of the latent space. The transformations are modeled by a collection of stacked neural components such as convolution network (Lecun et al. 1998). In this paper, we build a fully connected network for our CSC. The experiment studies will show that such a simple network can also achieve promising improvement upon well-established baseline methods.

For clarity, we start explanation on our model with the simplest case (*i.e.* 1-hidden layer network) as follows:

$$\mathbf{h}_i = g(\mathbf{x}_i|\mathbf{\Theta}) = g(\mathbf{W}\mathbf{x}_i + \mathbf{b}), \tag{2}$$

where $g(\cdot)$ is a nonlinear activation function, $\mathbf{\Theta} = \{\mathbf{W}, \mathbf{b}\}$ denotes the parametric network, $\mathbf{W} \in \mathcal{R}^{d \times m}$ denotes the weight, and $\mathbf{b} \in \mathcal{R}^d$ is the bias. To obtain a good initialization of $\mathbf{\Theta}$, we adopt self-supervised learning approach (Hinton and Salakhutdinov 2006). To be exact, we train an autoencoder by

$$\min_{\mathbf{\Theta}} \|\mathbf{X} - \hat{\mathbf{X}}\|_F, \tag{3}$$

where $\hat{\mathbf{X}}$ is the reconstruction of $\mathbf{X}$, *i.e.* the output of the autoencoder. Once the autoencoder converges, we use the learned weights of encoding module to initialize our CSC.

To perform clustering, we propose the following KL divergence based objective function:

$$\min \sum_{i \neq j} f(\mathcal{P}_i, \mathcal{P}_j) = \min_{\mathbf{\Theta}, \mathbf{\Omega}} \sum_{i \neq j} \mathcal{P}_j \log \frac{\mathcal{P}_j}{\mathcal{P}_i}, \tag{4}$$

where $\mathcal{P}_i$ and $\mathcal{P}_j$ are conditional distributions (probability maps) between $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_n]$ and $\mathbf{\Omega} =$

$[\omega_1, \omega_2, \cdots, \omega_k]$ in terms of two different metrics, and $\mathbf{\Omega}$ denotes cluster centers. Clearly, our objective function is proposed to achieve *invariance of distribution* by minimizing the discrepancy between the target distribution $\mathcal{P}_j$ and the predicted distribution $\mathcal{P}_i$. Noticed that, $\mathcal{P}_j$ is only used as a target and will not lead to the update of model according to the definition of KL divergence loss.

In this paper, we consider the binary-distribution case of eqn.(4) as below:

$$\min_{\mathbf{\Theta},\mathbf{\Omega}} \mathcal{P}_2 \log \frac{\mathcal{P}_2}{\mathcal{P}_1}, \tag{5}$$

where the final cluster assignment corresponds to the index of the maximal entry of $\mathcal{P}_1$.

Let $\mathcal{P}(\mathbf{h}_i|\omega_j)$ be the entry of $\mathcal{P}(\mathbf{H}|\mathbf{\Omega})$, *i.e.* the conditional distribution of $\mathbf{h}_i$ *w.r.t.* ω_j, we give its definition by:

$$\mathcal{P}(\mathbf{h}_i|\omega_j) = \frac{\mathcal{Q}^2(\mathbf{h}_i|\omega_j)/f_j}{\sum_j \mathcal{Q}^2(\mathbf{h}_i|\omega_j)/f_j}, \tag{6}$$

where $\mathcal{Q}(\mathbf{h}_i|\omega_j)$ denotes the closeness between $\mathbf{h}_i$ and ω_j, and f_j is the frequency for each cluster which is used to normalize loss contribution of each center, *i.e.* to prevent larger clusters from distorting the hidden space (Xie, Girshick, and Farhadi 2016).

Many existing distance metrics can be used to define $\mathcal{Q}(\mathbf{h}_i|\omega_j)$. In this paper, we adopt two most widely-used, *i.e.* Euclidean and Cosine distance. Specifically, we have

$$\mathcal{Q}_1(\mathbf{h}_i|\omega_j) = \max(0, \mu_i - z_{ij}), \tag{7}$$

where $z_{ij} = \|\mathbf{h}_i - \omega_j\|_2$ is the Euclidean distance between the i-th data and the j-th cluster centroid. μ_i is the mean of $\mathbf{z}_i$, *i.e.* the average distance between the i-th sample to all the cluster centroids. With eqn.(7), the Euclidean distance based dissimilarity is transformed into similarity, and meanwhile guaranteeing sparsity to the distribution (Coates, Lee, and Ng 2011). Another distribution $\mathcal{Q}_2$ is defined as the reciprocal of Cosine distance. In mathematic,

$$\mathcal{Q}_2(\mathbf{h}_i|\omega_j) = (2 - \frac{\mathbf{h}_i \cdot \omega_j}{\|\mathbf{h}_i\|_2 \|\omega_j\|_2})^{-1}, \tag{8}$$

noticed that, we use the constant with value of 2 instead of 1 to avoid trivial solutions.

It should be pointed out that, our clustering-oriented verification is different from traditional verification. The traditional one aims to judge whether a pair of samples comes from the same subject, whereas our formulation aims to minimize the difference between two distributions based on different distance metrics. More specifically, the traditional verification involves negative pairs (samples from different subjects) as well as positive pairs, whereas CSC only considers positive case since the distributions are modeled based on the same data point and cluster centroids. In fact, we assume that the performance of CSC could be further improved by considering negative pair since it would brings more diversity into our model. However, this is beyond the scope of this work and may be further explored in future.

Table 1: The used parameters of CSC. l denotes learning rate which is divided by de for each ep epochs and this operation is repeated $re - 1$ times unless convergence. bs denotes the batch size.

data sets	d	lr	de	ep	re	bs
mnist-full	10	10^{-5}	0.9	300	15	256
mnist-test	10	10^{-5}	0.5	$1,000$	3	256
reuters	10	1.0	0.9	500	6	256
cifar10	100	10^{-4}	0.9	300	10	256

Implementation Details

We optimize CSC using stochastic sub-gradient descent (SGD) with momentum and weights decay.

To initialize CSC, we train an m-500-500-2000-d-2000-500-500-m denoising autoencoder (Vincent et al. 2010) with a corruption ratio of 0.3, a momentum of 0.9, and a weight decay rate of 10^{-6}, where m and d are the dimension of input and feature space, respectively. Moreover, we adopt rectifier linear units (ReLu) (Glorot, Bordes, and Bengio 2011) as the activation function. Once the autoencoder converges, we use the weights of first four hidden layers to initialize $\mathbf{\Theta}$ and the cluster centers identified by kmeans to initialize $\mathbf{\Omega}$.

Experimental Results

In this section, we report the performance of our CSC for subspace clustering and compare it with 11 state-of-the-art approaches. For comprehensive studies, we adopt four metrics to evaluate the clustering quality.

Experiment Settings

For the proposed CSC, we implement it in Theano (Theano Development Team 2016) based Keras (Chollet 2015) which is a modular neural networks library. For baseline algorithms, we obtain the MATLAB codes from corresponding authors' websites. The experiments are conducted on a machine with a Titan X GPU and 24x Intel Xeon CPU.

Baseline Algorithms: We compare CSC with 10 clustering methods including kmeans, locality preserving nonnegative matrix factorization (NMF-LP) (Cai et al. 2009), Zeta function based agglomerative clustering (ZAC) (Zhao and Tang 2009), agglomerative clustering with average linkage (ACAL) (Jain, Murty, and Flynn 1999) and weighted linkage (ACWL) (Jain, Murty, and Flynn 1999), standard spectral clustering (SC) (Ng, Jordan, and Weiss 2001), LRR (Liu et al. 2013), low rank subspace clustering (LRSC) (Vidal and Favaro 2014), SSC (Elhamifar and Vidal 2013), and smooth representation clustering (SMR) (Hu et al. 2014). Moreover, we also report the results of our CSC without backpropagation which is identical to kmeans with denoising autoencoder (DAE+kmeans). For each algorithm, we tune their parameters for different data sets and then report their best performance. For our CSC, we only tune the parameters of SGD to guarantee convergence. The used parameters of CSC are summarized in Table 1.

Table 2: Performance comparisons with 11 clustering approaches. The **bold** numbers indicate the best results, and *Pars* reports the tuned parameters for the evaluated algorithms, *i.e.* ZAC (K, a, z), LRR (λ), LRSC (λ), NMF-LP (α), SC (α), SMR (α, ϵ), SSC (λ, ϵ).

Data Set	mnist-full					mnist-test				
Methods	Accuracy	NMI	ARI	Precision	Pars	Accuracy	NMI	ARI	Precision	Pars
kmeans	55.27%	52.74%	40.28%	44.99%	-	56.07%	53.58%	41.54%	46.41%	-
NMF-LP	48.85%	42.63%	29.89%	36.39%	7.0	66.48%	46.40%	39.84%	59.99%	5
ZAC	60.00%	65.47%	54.07%	44.35%	20,0.95,0.02	60.16%	66.01%	54.30%	44.42%	60,0.95,0.01
ACAL	11.77%	0.51%	0.02%	10.04%	-	12.20%	0.81%	0.00%	10.03%	-
ACWL	30.83%	22.31%	11.59%	17.03%	-	41.58%	38.96%	26.18%	28.46%	-
LRR	11.07%	0.43%	0.03%	10.01%	0.01	59.27%	59.16%	46.29%	47.07%	0.01
LRSC	13.67%	0.98%	0.26%	10.16%	0.05	60.21%	53.90%	43.14%	47.12%	0.03
SC	71.28%	73.18%	62.18%	62.58%	1.0	69.33%	70.97%	59.75%	60.43%	1.0
SMR	22.89%	35.74%	9.78%	41.81%	2^{-14},10^{-3}	67.59%	41.33%	36.23%	56.35%	2^{-16},0.01
SSC	67.65%	69.37%	58.61%	60.36%	0.01,0.01	60.96%	64.65%	50.79%	50.28%	0.01,0.01
DAE+kmeans	74.15%	69.70%	63.90%	65.44%	-	85.20%	72.07%	70.47%	72.87%	-
CSC	**87.16%**	**75.50%**	**74.27%**	**76.43%**	-	**86.49%**	**73.34%**	**72.88%**	**75.37%**	-

Table 3: Performance comparisons with 11 clustering approaches. The **bold** numbers indicate the best results, and *Pars* reports the tuned parameters for the evaluated algorithms, *i.e.* ZAC (K, a, z), LRR (λ), LRSC (λ), NMF-LP (α), SC (α), SMR (α, ϵ), SSC (λ, ϵ).

Data Set	cifar10					reuters				
Methods	Accuracy	NMI	ARI	Precision	Pars	Accuracy	NMI	ARI	Precision	Pars
kmeans	19.88%	6.39%	3.09%	12.67%	-	54.01%	34.91%	27.79%	45.51%	-
NMF-LP	17.97%	5.10%	2.58%	12.26%	10	66.48%	34.40%	39.84%	59.99%	5
ZAC	10.14%	0.17%	0.00%	9.99%	20,0.95,0.01	43.66%	1.11%	0.71%	31.38%	20,0.95,0.01
ACAL	10.90%	0.51%	0.03%	10.00%	-	43.98%	0.16%	0.06%	31.48%	-
ACWL	15.79%	3.62%	1.87%	11.01%	-	42.17%	0.75%	-0.65%	30.90%	-
LRR	11.07%	0.43%	0.03%	10.01%	0.01	53.94%	33.42%	27.35%	45.54%	0.1
LRSC	20.79%	6.31%	3.81%	12.23%	0.05	64.26%	32.33%	32.01%	51.97%	10^{-4}
SC	20.18%	6.73%	3.33%	12.83%	10	66.33%	33.91%	30.40%	48.69%	50
SMR	20.76%	6.29%	3.81%	12.13%	2^{-16},0.1	67.59%	34.33%	36.23%	56.35%	2^{-16},0.02
SSC	19.82%	6.10%	3.25%	12.84%	0.01,0.01	43.22%	0.21%	0.07%	31.15%	0.01,0.01
DAE+kmeans	20.86%	6.95%	3.67%	12.94%	-	63.96%	35.81%	38.63%	61.52%	-
CSC	**21.88%**	**7.01%**	**3.90%**	**13.29%**	-	**69.72%**	**35.98%**	**44.45%**	**62.12%**	-

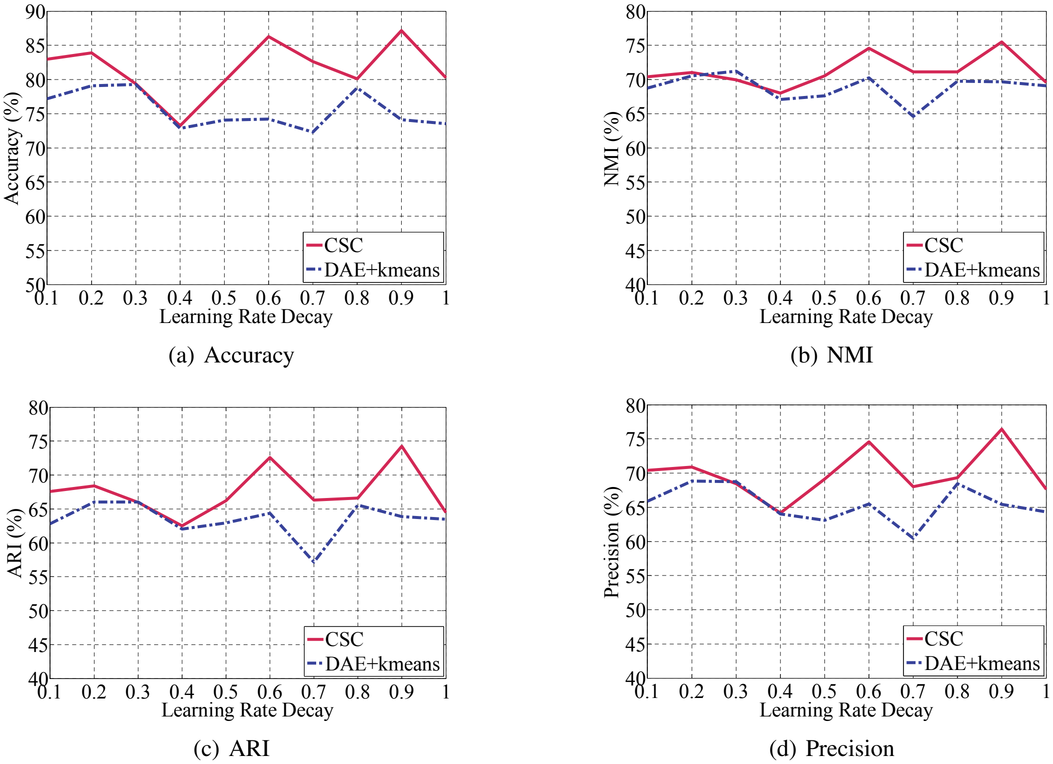
Data Sets: We use four data sets for our experiments, *i.e.* full mnist data set (Lecun et al. 1998) (mnist-full), the test partition of mnist (mnist-test), the testing subset of cifar10 (Krizhevsky and Hinton 2009), and a subset of reuters (Lewis et al. 2004). mnist-full and mnist-test consist of 70,000 and 10,000 28 × 28 images, respectively. All mnist images are distributed over 10 handwritten digits. The cifar10 testing partition includes 10,000 32 × 32 images that are sampled from 10 objects. The used reuters data set includes 10,000 documents from four root categories and each document is represented as a term-frequency-inverse-document-frequency feature vector with 2,000 most frequently words. For all the used data sets, we do not perform any pre-processing steps excepted centering each sample to their centroids and truncating over-high values resulted from noises.

Evaluation Criteria: Four popular metrics are used to evaluate the clustering quality, *i.e. Accuracy*, normalized mutual information (*NMI*), adjusted rand index (*ARI*), and *Precision*. Higher value of these metrics indicates better performance.

Comparisons with State-of-the-art Methods

We first compare our method with some popular clustering algorithms on four data sets. Tables 2–3 report the results from which we observe that:

- Our CSC method achieves remarkable improvements comparing with 11 clustering approaches. For example, on the mnist-full data set, the *Accuracy* of CSC is 15.88% at least higher than that of the best baseline approach. Specifically, 87.16% of CSC versus 71.28% of SC.
- In terms of other three evaluation metrics, CSC is also the best algorithm on all the used data sets. For example, it is 2.37%, 13.13%, and 14.94% higher than the other algorithms on mnist-full regarding to *NMI*, *ARI*, and *Precision*, respectively.
- The results of DAE+kmeans and CSC show the effectiveness of our method. Considering mnist-full, for example, the performance gains in *Accuracy*, *NMI*, *ARI*, and *Precision* of CSC over DAE+kmeans are 13%, 6%, 10%, and 9%, respectively.
- On the small scale data set (*e.g.* mnist-test), although existing clustering methods are inferior to our deep model,



(a) Accuracy

(b) NMI

(c) ARI

(d) Precision

Figure 3: The influence of the learning rate decay of SGD. For every 300 epochs, the learning rate is reduced by multiplying with the decay ratio.

but the performance of these approaches is still acceptable. When a larger scale data set is used (*e.g.* mnist-full), almost all these approaches are failed to achieve a desirable result. In contrast, our CSC still performs stable and separates most data point into corrected clusters.

Influence of Different Parameters

One of major challenges of deep neural network is requiring tuning various parameters, which is an exhausted task. In this section, we investigate the influence of user-specified parameters. In evaluations, we also report the performance of kmeans with denoising autoencoder, denoted by DAE+kmeans. All experiments are conducted on the full mnist data set and all parameters excepted the evaluated one are fixed as shown in Table 1.

The choice of learning rate directly decides whether our model converges, but we experimentally found that learning rate decay plays a more important role than learning rate. Thus, we examine the influence of learning rate decay instead of learning rate in this section. The result is demonstrated in Figure 3. From results, we could see that CSC achieves a reasonable fluctuation when the learning rate decay is larger than 0.6 in terms of four performance metrics.

Conclusion

In this paper, we proposed reformulating subspace clustering as a verification problem by minimizing the discrepancy between pairwise sample-centers distributions. To verify the effectiveness of our idea, we designed a fully connected neural network based subspace clustering method, termed CSC. CSC jointly learns a collection of hierarchical representation and cluster assignment in end-to-end manner. In future, we plan to investigate the performance of our method when other modules such as convolution neural network are used. Moreover, it is also interesting to further improve the performance of CSC using multiple metrics learning and incorporating contrastive divergence with negative pairs.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of this paper. X. Peng was supported by A*STAR Industrial Robotics Programme - Distributed Sensing and Perception under SERC grant 1225100002. J. Feng was partially supported by NUS startup grant R263000C08133 and Ministry of Education of Singapore AcRF Tier One grant R-263-000-C21-112. J. Lu was supported by the National Key Research and Development Program of China under Grant 2016YFB1001001, the NFSC under Grant 61672306, and the National 1000 Young Talents Plan Program. Z. Yi was supported by National Nature Science Foundation of China under grant No. 61432012.

References

Belkin, M., and Niyogi, P. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.* 15(6):1373–1396.

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35(8):1798–1828.

Cai, D.; He, X.; Wang, X.; Bao, H.; and Han, J. 2009. Locality preserving nonnegative matrix factorization. In *Proc. of 21st Int. Jnt Conf on Artif Intell*, 1010–1015. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Chollet, F. 2015. Keras. https://github.com/fchollet/keras.

Coates, A.; Lee, H.; and Ng, A. 2011. An analysis of single-layer networks in unsupervised feature learning. In *Proc. of 14th Int. Conf. on Artif Intell and Stat*, volume 15, 215–223. Ft. Lauderdale, FL: JMLR W&CP.

Elhamifar, E., and Vidal, R. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.* 35(11):2765–2781.

Feng, J.; Lin, Z. C.; Xu, H.; and Yan, S. C. 2014. Robust subspace segmentation with block-diagonal prior. In *Proc. of 27th IEEE Conf. Comput. Vis. and Pattern Recognit.*, 3818–3825.

Glorot, X.; Bordes, A.; and Bengio, Y. 2011. Deep sparse rectifier neural networks. In *Proc. of 14th Int. Conf. on Artif Intell and Stat*, volume 15, 315–323. Ft. Lauderdale, FL: JMLR W&CP.

Hinton, G. E., and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *Science* 313(5786):504–507.

Hu, H.; Lin, Z. C.; Feng, J.; and Zhou, J. 2014. Smooth representation clustering. In *Proc. of 27th IEEE Conf. Comput. Vis. and Pattern Recognit.*, 3834–3841.

Hu, J. L.; Lu, J. W.; and Tan, Y. P. 2014. Discriminative deep metric learning for face verification in the wild. In *Proc. of 27th IEEE Conf. Comput. Vis. and Pattern Recognit.*, 1875–1882.

Jain, A. K.; Murty, M. N.; and Flynn, P. J. 1999. Data clustering: A review. *ACM Comput. Surv.* 31(3):264–323.

Krizhevsky, A., and Hinton, G. 2009. Learning multiple layers of features from tiny images.

Krizhevsky, A.; Sutskever, L.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Proc. of 25th Adv. in Neural Inf. Process. Syst.*, 1097–1105.

Lecun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *Nature* 521(7553):436–444.

Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proc. of IEEE* 86(11):2278–2324.

Lewis, D. D.; Yang, Y.; Rose, T. G.; and Li, F. 2004. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.* 5:361–397.

Liu, G. C.; Lin, Z. C.; Yan, S. C.; Sun, J.; Yu, Y.; and Ma, Y. 2013. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.* 35(1):171–184.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. In *Proc. of 14th Adv. in Neural Inf. Process. Syst.*, 849–856.

Peng, X.; Lu, C.; Zhang, Y.; and Tang, H. 2016a. Connections between nuclear norm and frobenius norm based representation. *IEEE Trans Neural Netw. Learn. Syst.* PP(99):1–7.

Peng, X.; Xiao, S.; Feng, J.; Yau, W.; and Yi, Z. 2016b. Deep subspace clustering with sparsity prior. In *Proc. of 25th Int. Jt. Conf. Artif. Intell.*, 1925–1931.

Peng, X.; Yu, Z.; Yi, Z.; and Tang, H. 2016c. Constructing the l2-graph for robust subspace learning and subspace clustering. *IEEE Trans. Cybern.* PP(99):1–14.

Peng, X.; Yi, Z.; and Tang, H. J. 2015. Robust subspace clustering via thresholding ridge regression. In *Proc. of 29th AAAI Conf. Artif. Intell.*, 3827–3833.

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints* abs/1605.02688.

Vidal, R., and Favaro, P. 2014. Low rank subspace clustering (LRSC). *Pattern Recognit. Lett.* 43:47 – 61.

Vidal, R. 2011. Subspace clustering. *IEEE Signal Proc. Mag.* 28(2):52–68.

Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; and Manzagol, P.-A. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* 11:3371–3408.

Wang, Q.; Lin, J.; and Yuan, Y. 2016. Salient band selection for hyperspectral image classification via manifold ranking. *IEEE Trans. Neural Netw. Learn. Syst.* 27(6):1279–1289.

Wang, Q.; Zhu, G.; and Yuan, Y. 2014. Statistical quantization for similarity search. *Comput. Vis. Image Underst.* 124:22–30. Large Scale Multimedia Semantic Indexing.

Xiao, S.; Tan, M.; Xu, D.; and Dong, Z. Y. 2015. Robust kernel low-rank representation. *IEEE Trans. Neural. Netw. Learn. Syst.* PP(99):1–1.

Xie, J. Y.; Girshick, R.; and Farhadi, A. 2016. Unsupervised deep embedding for clustering analysis. In *Proc. of 33th Int. Conf. Mach. Learn.*

Yang, J.; Parikh, D.; and Batra, D. 2016. Joint unsupervised learning of deep representations and image clusters. In *Proc. of 29th IEEE Conf. Comput. Vis. and Pattern Recognit.*

Yu, Z. D.; Liu, W. Y.; Liu, W.; Peng, X.; Hui, Z.; and Kumar, B. V. 2015. Generalized transitive distance with minimum spanning random forest. In *Proc. of 24th Int. Joint Conf. on Artif. Intell.*, 2205–2211.

Zhao, D., and Tang, X. 2009. Cyclizing clusters via zeta function of a graph. In *Proc. of 21th Adv. in Neural Inf. Process. Syst.*, 1953–1960.