Deep Self-Evolution Clustering

Jianlong Chang, Gaofeng Meng, Senior Member, IEEE, Lingfeng Wang, Member, IEEE, Shiming Xiang, Member, IEEE, and Chunhong Pan, Member, IEEE

Abstract—Clustering is a crucial but challenging task in pattern analysis and machine learning. Existing methods often ignore the combination between representation learning and clustering. To tackle this problem, we reconsider the clustering task from its definition to develop Deep Self-Evolution Clustering (DSEC) to jointly learn representations and cluster data. For this purpose, the clustering task is recast as a binary pairwise-classification problem to estimate whether pairwise patterns are similar. Specifically, similarities between pairwise patterns are defined by the dot product between indicator features which are generated by a deep neural network (DNN). To learn informative representations for clustering, clustering constraints are imposed on the indicator features to represent specific concepts with specific representations. Since the ground-truth similarities are unavailable in clustering, an alternating iterative algorithm called Self-Evolution Clustering Training (SECT) is presented to select similar and dissimilar pairwise patterns and to train the DNN alternately. Consequently, the indicator features tend to be one-hot vectors and the patterns can be clustered by locating the largest response of the learned indicator features. Extensive experiments strongly evidence that DSEC outperforms current models on twelve popular image, text and audio datasets consistently.

Index Terms—Clustering, deep self-evolution clustering, self-evolution clustering training, deep unsupervised learning

1 INTRODUCTION

CLUSTERING, an essential data analysis tool in pattern analysis and machine learning, is a data-driven task that attempts to explore knowledge from unlabeled data purely. In a multitude of science and engineering fields where the label information is unobservable or costly to obtain, many applications can be considered as typical instances of clustering, such as the astronomical data analysis [1], the medical analysis [2], the gene sequencing [3], and the information retrieval [4], [5], [6], [7].

In the literature, much research has been devoted to clustering data [8], [9], [10], [11], [12], [13], [14], [15]. From the definition, clustering produces an effective organization of data based on similarities, namely patterns within the same clusters are similar to each other, while those in different clusters are dissimilar. Technically, there are two essential issues that require to be handled in clustering, i.e., how to choose a right distance metric to measure the similarities and how to define a justified clustering pattern to describe data? Such issues originate from that the objective of clustering is not clear enough. Generally, traditional clustering methods handle the issues by introducing additional assumptions, i.e., measuring the similarities via predetermined distance metrics and clustering data based on predefined clustering patterns.

To estimate similarities, plentiful attention has been paid to predetermine proper distance metrics. Distance, generally, depends on feature spaces of data. That is, a crucial problem of estimating similarities is to acquire informative representations of data in an unsupervised way. Based on this insight, many attempts have dedicated to developing unsupervised representation extracting techniques to measure similarities in latent spaces. Early methods usually adopt manually designed feature descriptors, including HOG [16] and SIFT [17]. While these feature descriptors may loose impacts from some messy variables (e.g., rotation, translation, etc.), they often suffer from appearance variations of scenes and objects. To overcome the limitation, many deep unsupervised learning methods are proposed recently, such as the auto-encoder [18], the denoising autoencoder [19], and the auto-encoding variational bayes [20]. Technically, these methods aim to recode patterns by using representations that are capable of reconstructing inputs. Despite the observable advances, there is still much room for improvement, e.g., learning high-level representations with high interpretability.

To predefine clustering patterns, lots of efforts have been widely made. Traditional methods, in principle, are devoted to taking a comprehensive view of data from diverse aspects [21]. Among these methods, the Gaussian mixture distribution is a popular descriptor to describe clustering patterns. Based on this perspective, a series of clustering methods are proposed, including the K-means [11] and the K-means++ [22]. Contrary to predefining a distribution, conceivable methods prefer to estimate proper density functions to predefine clustering patterns, such as the density-based spatial clustering [23] and the density peaks based model [24].
In spite of their primitively impressive successes, a serious issue related to these methods is that the performance severely relies on the predefined clustering patterns, which are difficult to define before clustering.

Although the achievements in the literature are brilliant, traditional methods still suffer from a multi-stage clustering paradigm, i.e., extracting representations of data with unsupervised methods and clustering data depending on the representations. From the viewpoint of applications, the multi-stage clustering paradigm is cumbersome in practice. From the viewpoint of performance, the generated representations are fixed after the unsupervised extraction. As a result, the representations can not be further improved to achieve better performance during the clustering process. To eliminate this problem, the primal discriminative methods [25], [26] that alternate between clustering and feature learning are proposed. However, these models [25], [26] sole pertain to learn low-level representations with shallow methods, such as linear functions [25] and kernel functions [26].

To learn high-level representations and cluster data simultaneously, in this paper, we develop Deep Self-Evolution Clustering (DSEC) that recasts the clustering task into a binary pairwise-classification problem to judge whether pairwise patterns belong to the same clusters. Specifically, similarities between pairwise patterns are calculated as dot product between indicator features generated by a deep neural network (DNN). For clustering, clustering constraints are introduced into the DSEC model to learn specific representations for specific concepts. Since the ground-truth similarities are unobservable in clustering, Self-Evolution Clustering Training (SECT) is presented to select similar and dissimilar pairwise patterns and to train the DNN alternately. As a result, the indicator features tend to be one-hot vectors and the patterns can be clustered by locating the largest response of the learned indicator features. Visually, an intuitive description about the motivation of DSEC is illustrated in Fig. 1.

To sum up, the main contributions of this work are:

- Starting from the definition, a binary pairwise-classification framework is developed for clustering, which explores an effective scheme from a distinctive viewpoint. The successful attempt signifies that considering relations between patterns is important for clustering, which offers an alternative orientation for unsupervised learning.
- By introducing a constraint into DSEC, the learned indicator features theoretically tend to be one-hot vectors where a specific one-hot vector corresponds to a specific cluster. Thus, DSEC can cluster data by locating the largest response of the learned indicator features purely, which dramatically simplifies the labeling process in clustering.
- A self-evolution training algorithm called SECT is proposed to optimize the DSEC model, which can gradually elevate the clustering performance by itself. Furthermore, the algorithm can effectively explore the supervised information to deal with clustering, which builds a bridge between the supervised and unsupervised learning seamlessly.

It should be noted that a preliminary version of this work was published in ICCV 2017 as an oral presentation [27]. In this paper, we further extend our previous work from both theoretical and empirical aspects. Theoretically, a more general clustering constraint is developed. Under this general constraint, the learned indicator features are one-hot vectors ideally, which is theoretically proved in this paper. Note that the previous work can just be regarded as a special case when an additional constraint is added to this general constraint. Furthermore, a clustering priori knowledge is introduced to guide DSEC to select similar and dissimilar pairwise patterns at the initial stage. Empirically, extensive experiments are conducted to demonstrate the effectiveness of DSEC as follows. First, we extend the preliminary version from the image clustering task to the text and audio clustering tasks, which indicates that our approach is effective for managing general clustering tasks in practice. Second, we comprehensively evaluate the performance of DSEC by making comparisons with more recent methods, presenting more visualization results, analysing results with various clustering scenarios, combing it with semi-supervised tasks, and so on.

The remainder of this paper is organized as follows: in Section 2, we make a brief review on the related work of clustering, deep learning and deep unsupervised learning. Sections 3 and 4 describe the details of the developed DSEC model and DSEC algorithm, respectively. Comprehensive experimental results are reported and analyzed in Section 5. Finally, Section 6 concludes this paper.

2 RELATED WORK

In this section, we will make a brief review of the related work on the clustering, deep learning and deep unsupervised learning methods.

2.1 Clustering

From the aspect related to algorithmic structure and operation, clustering methods can be roughly divided into two categories: agglomerative and divisive methods [21].

The agglomerative approaches start with each pattern regarded as a diverse cluster and merge clusters together gradually until stopping criterions are met. Proceed from this definition, some essential approaches are presented, e.g., the agglomerative methods [8], [28] and the hierarchical clustering method [29]. By combining the agglomerative
One approach [8] with deep learning, recently, joint unsupervised learning is presented for representation learning and clustering simultaneously [30]. Contrary to agglomerative approaches, divisive methods attempt to split all patterns concurrently, until stopping criteria are satisfied. The frequently used methods are the K-means [8] and its variants, such as the constrained K-means [11] and the K-means++ [22]. From the perspective of graph theory, spectral clustering models divide patterns into diverse clusters if they are poorly connected [31]. Furthermore, these insights also form the bases of a number of methods, e.g., the density-based clustering method [24].

Despite the competitive successes in clustering, these traditional methods are typically developed for simple and monotonic application scenarios, and may fail in more complex environments, such as large scale images, texts, and speeches clustering. The limitation inherits from that the methods perform clustering based on the predefined representations, which implies that the representations can not be further improved to achieve better performance dynamically.

2.2 Deep Learning

Deep learning, also termed as deep neural network, is a significant technique in pattern analysis and machine learning, which has been developed to manage tasks like humans do [32]. Specifically, the pivotal advance of DNNs is that they allow machines to adaptively learn informative representations of data to tackle specific tasks via a unified training procedure, without any prior knowledge or interventions of humans.

Although the birth of deep learning can be traced to 1960s, much interesting research has been proposed in the past decade. This is because there were some limitations in the past, including computing resource and data size. With the development of technology and the accumulation of knowledge, by applying two efficient GPUs, Krizhevsky et al. won the championship of the ImageNet Large Scale Visual Recognition Competition in 2012 (ILSVRC-2012) based on a DNN named as AlexNet [33], which triggered a wave of deep learning. Followed that, various deep DNN architectures are exploited, e.g., VGG [34], GoogleNet [35] and ResNet [36]. Thanks the extraordinary capability of DNNs in representation learning, significant successes have been achieved in various pattern analysis and machine learning problems [37], especially in computer vision, such as image classification [34], [35], [36], image segmentation [38], [39], [40], object detection [41], [42].

Although the achievements are remarkable, deep learning methods suffer from a critical limitation that a large amount of labeled data is required to learn parameters. Such limitation comes from that numerous parameters in deep models require to be estimated with a large amount of labeled data. However, it is laborious and expensive to collect sufficient labeled data. Especially, labeled data may not be previously acquired in many practical tasks, e.g., information retrieval.

2.3 Deep Unsupervised Learning

To overcome the limitations in the supervised learning, deep unsupervised learning has drawn much attention to train DNNs in an unsupervised manner. Based on the unlabeled data purely, as a result, DNNs can be trained to some justified status, which can be used to extract convicive representations for various unsupervised tasks, including clustering [43] and retrieval [44], [45].

Over the last decade, many deep unsupervised learning methods have been explored for learning informative representations or clustering. Primitively, most unsupervised methods aim to pre-train DNNs by reconstructing inputs, such as the restricted Boltzmann machine [46], the autoencoder [18] and the denoising autoencoder [19]. These methods attempt to train DNNs to learn non-linear functions that are capable of reconstructing inputs. In order to learn more robust representations, several popular regularization terms have been introduced, e.g., the sparsity constraint [47]. To simplify the learning process, consistent inference of latent representations based learning method omits the decoding part to learn more interpretive representations directly [48]. Recently, deep generative models, including the auto-encoding variational bayes [20] and the generative adversarial network [49], have been proposed to encode data. Technically, these generative models devote to encoding data distributions with DNNs. By introducing some prior distribution to learn hidden representations, several deep unsupervised learning methods have been developed, such as the adversarial autoencoder [50], the categorical generative adversarial networks [51] and the Gaussian mixture variational autoencoder [52].

While much endeavor has been devoted to deep unsupervised learning, several intrinsic challenges still remain. First, how to provide proper driving force to train DNNs in an unsupervised manner? Second, clustering results can not be obtained directly via the learned feature representations by these methods, which implies that the generated representations are unsuitable for clustering.

3 DSEC Model

Clustering, technically, is a process of grouping data into clusters such that patterns within the same clusters are similar to each other, while those in different clusters are dissimilar. Therefore, pairwise patterns only belong to either the same clusters or different clusters. Based on this perspective, DSEC recasts the clustering task into a binary pairwise-classification framework. Specifically, the flowchart of DSEC is illustrated in Fig. 2 and more details are given in the following subsections.

3.1 Binary Pairwise-Classification for Clustering

Given an unlabeled dataset \( \mathcal{X} = \{ \mathbf{x}_i \}_{i=1}^n \) and the predefined number of clusters \( k \), where \( \mathbf{x}_i \) indicates the \( i \)th pattern, DSEC manages the clustering task by investigating similarities between pairwise patterns. To consider the similarities, we redefine the dataset as \( \mathcal{D} = \{ (\mathbf{x}_i, \mathbf{x}_j, r_{ij}) \}_{i,j=1,i\neq j}^n \), where \( \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X} \) are the unlabeled patterns (which refer to the inputs) and \( r_{ij} \in \mathcal{Y} \) is an unobservable binary variable (which refers to the output). Specifically, \( r_{ij} = 1 \) implies that \( \mathbf{x}_i \) and \( \mathbf{x}_j \) belong to the same cluster, and \( r_{ij} = 0 \) otherwise. Formally, the objective function of DSEC is formulated as follows:

\[
\min_{w} L(w) = \sum_{i=1}^{n} \sum_{j=1}^{n} L(r_{ij}, g(\mathbf{x}_i, \mathbf{x}_j; w)),
\]
where $L(r_{ij}, g(x_i, x_j; \mathbf{w}))$ measures the loss between the binary variable $r_{ij}$ and the estimated similarity $g(x_i, x_j; \mathbf{w})$, and $\mathbf{w}$ signifies the model parameters in the function $g$. Specifically, the squared error loss is employed as $L(r_{ij}, g(x_i, x_j; \mathbf{w})) = \|r_{ij} - g(x_i, x_j; \mathbf{w})\|^2$. (2)

Generally, two essential issues need to be tackled in DSEC. First, assuming that $\mathcal{Y}$ is observable, the clustering labels of $x_i$ and $x_j$ are still unacquirable by only accessing to the estimated similarity $g(x_i, x_j; \mathbf{w})$. Second, the ground-truth values in $\mathcal{Y}$ cannot be observed in clustering. In the following, Sections 3.2 and 3.3 focus on tackling the aforementioned two issues, respectively.

### 3.2 Indicator Features under Clustering Constraints

To learn more informative representations for clustering, indicator features are introduced into the DSEC model to indicate clustering labels, namely representing a concept with a specific neuron. Denote the indicator features as $\mathcal{I} = \{\mathbf{I}_i\}_{i=1}^n$, where $\mathbf{I}_i$ is a $k$-dimensional representation of the pattern $x_i$, a clustering constraint $\mathcal{C}$ is imposed on the indicator features as follows:

$$\mathcal{C} : \forall i, \mathbf{I}_{ih} \geq 0, h = 1, \ldots, k,$$  (3)

where $I_{ih}$ represents the $h^{th}$ element of the indicator feature $\mathbf{I}_i$. For clarity, “$\mathcal{C}(\mathbf{I}_i) = \text{True}$” is utilized to describe the case that $\mathbf{I}_i$ satisfies the clustering constraint $\mathcal{C}$. Because of the brilliant capability of function fitting, DNNs are employed to learn the indicator features, i.e.,

$$\mathbf{I}_i = f(x_i; \mathbf{w}),$$  (4)

where $f(\cdot; \mathbf{w})$ indicates a DNN model that is uniquely determined by the parameter $\mathbf{w}$.

DSEC measures the similarities via dot product between the indicator features. Formally, the similarities between patterns is written as follows:

$$g(x_i, x_j; \mathbf{w}) = f(x_i; \mathbf{w}) \cdot f(x_j; \mathbf{w}) = \mathbf{I}_i \cdot \mathbf{I}_j,$$  (5)

where the operator “$\cdot$” represents dot product between two vectors. By introducing the indicator features, the DSEC model is reformulated as:

$$\min_{\mathbf{w}} E(\mathbf{w}) = \sum_{i=1}^n \sum_{j=1}^n L(r_{ij}, \mathbf{I}_i \cdot \mathbf{I}_j),$$  (6)

s.t. $\forall i$, $\mathbf{I}_i = f(x_i; \mathbf{w})$, $\mathcal{C}(\mathbf{I}_i) = \text{True}$.

The clustering constraint in Eq. (3) can bring a dramatically attractive property for clustering. Let $\mathbb{E}^k$ denote the standard basis of the $k$-dimensional Euclidean space, we have the following theorem (the proof of this theorem is reported in the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieee-computersociety.org/10.1109/TPAMI.2018.2889949):

**Theorem 1.** If the optimal value of Eq. (6) is attained, for $\forall i$,

$$\mathbf{I}_i \in \mathbb{E}^k,$$

for $\forall i$ and $\forall j$

$$\mathbf{I}_i \neq \mathbf{I}_j \Leftrightarrow r_{ij} = 0 \text{ and } \mathbf{I}_i = \mathbf{I}_j \Leftrightarrow r_{ij} = 1.$$  

Theorem 1 signifies that the learned indicator features are $k$-dimensional one-hot vectors ideally. Specifically, the activated neurons are identical if pairwise patterns belong to the same cluster and are different otherwise, which is similar to the behaviour of grandmother cells in brain [53]. Benefited from this property, DSEC is capable of clustering based on the learned indicator features only.

1. The grandmother cell is a hypothetical neuron that represents a complex but specific concept or object.
where $p \in (0, +\infty)$ is satisfied. For clarity, we refer to this clustering constraint as $C_p$. Specifically, the search spaces of the 2-dimensional indicator features under diverse clustering constraints are depicted in Fig. 3. From $C$ to $C_p$, visually, the search spaces of the indicator features will be reduced from a plane (i.e., the first quadrant) to curves that still include the optimal indicator features (i.e., $[1, 0]$ and $[0, 1]$). That is, for arbitrary $p \in (0, +\infty)$, Theorem 1 is met under the arbitrary constraint $C_p$. Since the search spaces are significantly reduced, further, higher efficiency will be acquired by $C_p$, which will be empirically demonstrated in the experimental section.

### 3.3 Labeled Pairwise Patterns Selection

Since $\mathcal{Y}$ is unobservable in clustering, we develop a self-evolution mechanism that allows DSEC to gradually select similar and dissimilar pairwise patterns to train the DNNs in DSEC. For this purpose, DSEC gradually selects labeled pairwise patterns with high likelihood with the progress of the learning, which is inspired from the curriculum learning [54] in the supervised learning. That is, “easy” pairwise patterns with high likelihood are first selected as training data to find rough clusters. Then, as the learning progresses, the trained DNNs can be utilized for measuring more accurate similarities and more pairwise patterns will be gradually selected to find more refined clusters.

In DSEC, labeled pairwise patterns are selected based on similarities between indicator features, which is inspired by the following two observations. First, if DNNs are already trained to some justified status with “easy” pairwise patterns, effective indicator features can be learned to estimate similarities between pairwise patterns. Second, for randomly initialized DNNs, they also can capture some available information of patterns [55], despite it is quite limited. According to the observations, similarities between patterns can be reflected by similarities between indicator features approximatively. Therefore, DSEC selects labeled pairwise patterns as follows:

$$
C_p : \forall i, \|I_i\|_p = 1, \quad I_{ih} \geq 0, \quad h = 1, \ldots, k,
$$

where $p \in (0, +\infty)$ is satisfied. For clarity, we refer to this clustering constraint as $C_p$. Specifically, the search spaces of the 2-dimensional indicator features under diverse clustering constraints are depicted in Fig. 3. From $C$ to $C_p$, visually, the search spaces of the indicator features will be reduced from a plane (i.e., the first quadrant) to curves that still include the optimal indicator features (i.e., $[1, 0]$ and $[0, 1]$). That is, for arbitrary $p \in (0, +\infty)$, Theorem 1 is met under the arbitrary constraint $C_p$. Since the search spaces are significantly reduced, further, higher efficiency will be acquired by $C_p$, which will be empirically demonstrated in the experimental section.

From Theorem 2, we have the following corollary,

**Corollary 1.** For the balanced dataset $\mathcal{X}$, $P = \frac{1}{2}$.

From the aforementioned theorem and corollary, we can acquire two important insights for clustering. Theorem 2 signifies that the prior probability of pairwise patterns belonging to different clusters is higher than to the same clusters generally. On the balanced datasets, shown in Corollary 1, pairwise patterns that belong to the same clusters account for only $1/k$, where $k$ is the number of clusters. From these insights, if most of pairwise patterns are labeled as dissimilar data, most of pairwise patterns can acquire the ground-truth similarities. As a result, informative gradient can be yielded to train randomly initialized DNNs to considerable status, which guarantee the availability of our method in practice.
4 DSEC Algorithm

In this section, a self-evolution optimization algorithm is developed to train the DSEC model in Eq. (9), and a label inference tactic is presented to cluster each pattern based on the corresponding indicator feature purely.

4.1 Self-Evolution Clustering Training

Akin to [25], [26], an alternating iterative optimization algorithm called Self-Evolution Clustering Training, is developed to optimize the DSEC model. The algorithm focuses on two issues, namely the implementation of the clustering constraints and the iterative optimization.

A constraint layer is devised to meet the requirements of the clustering constraints in Eq. (7). Formally, the functions in this layer is formulated as follows:

$$I^{\text{tem}}_h := \exp \left( I^n_h - \max_h \left( I^{\text{tem}}_h \right) \right), \quad h = 1, \cdots, k,$$

(11a)

$$I^{\text{out}}_h := \frac{I^{\text{tem}}_h}{\| I^{\text{tem}}_p \|_P}, \quad h = 1, \cdots, k,$$

(11b)

where $I^n$, $I^{\text{tem}}$ and $I^{\text{out}} \in \mathbb{R}^k$ indicate the input, intermediate variable and output of the constraint layer, orderly. Additionally, $I^n_h$, $I^{\text{tem}}_h$ and $I^{\text{out}}_h$ represent the $h$th element in $I^n$, $I^{\text{tem}}$ and $I^{\text{out}}$. Note that all elements of the output $I^{\text{out}}$ are mapped into $[0, +\infty)$ by Eq. (11a) and the output $I^{\text{out}}$ is simultaneously limited to meet the norm constraint by Eq. (11b). In DSEC, the DNNs are always followed by the constraint layer. That is, for arbitrary $i, \text{I}$, invariably satisfies the clustering constraints in Eq. (7).

The optimizations of $w$ and $\{u, l\}$ are alternating iterative performed. Once $\{u, l\}$ are fixed, $r$ and $v$ can be obtained, and the DSEC model degenerates as follows:

$$\min_w E(w) = \sum_{i=1}^{n} \sum_{j=1}^{n} v_{ij} L(r_{ij}, f(x_i; w) - f(x_j; w)).$$

(12)

Since $r$ and $v$ are available, optimizing Eq. (12) is a supervised learning problem so that the back-propagation learning algorithm can be utilized to optimize $w$. Besides, the storage complexity of $r$ and $v$ is $O(n^2)$ because the similarities between pairwise patterns need to be calculated and stored. It is too high to handle large datasets. To deal with this problem, $w$ is updated on each batch that is gradually sampled from the original dataset, as illustrated in the line 1 to line 1 of Algorithm 1.

Similarly, the DSEC model can be simplified as follows when $w$ is fixed,

$$\min_{l, u} E(l, u) = s(l, u).$$

(13)

According to the gradient descent algorithm, in each iteration, the update rules of $l$ and $u$ can be written as:

$$u := u - \eta \frac{\partial s(l, u)}{\partial u},$$

$$l := l - \eta \frac{\partial s(l, u)}{\partial l},$$

where $\eta > 0$ is the learning rate. Since $\frac{\partial s(l, u)}{\partial u} = -1 < 0$ and $\frac{\partial s(l, u)}{\partial l} = 1 > 0$ are always satisfied, $s(l, u)$ will gradually decreasing until $l = u$ is met. This corresponds to our target scenario that pairwise patterns are gradually appended for training with the progress of the optimization, until all pairwise patterns are considered.

Algorithm 1. Deep Self-Evolution Clustering

Input: Dataset $X = \{x_i\}_{i=1}^{n}$, $w$, $k$, $u$, $l$, $\eta$, $m$.
Output: Clustering label $c_i$ of $x_i \in X$.
1: Randomly initialize $w$;
2: repeat
3: for all $q \in \{1, 2, \cdots, \frac{n}{m}\}$ do
4: Sample batch $X_q$ from $X$; // $m$ patterns per batch
5: Select training data from $X_q$; // Eq. (8)
6: Calculate the indicator coefficient $v_i$; // Eq. (10)
7: Update $w$ by minimizing Eq. (12);
8: end for
9: Update $u$ and $l$ according to Eq. (14);
10: until $l > u$
11: for all $x_i \in X$ do
12: $l_i := f(x_i; w)$;
13: $c_i := \arg \max_h (I_{ih})$;
14: end for

4.2 Label Inference for Clustering

Clustering labels can be inferred via the learned indicator features purely, which benefits from the predictable modes of the learned indicator features that are $k$-dimensional one-hot vectors ideally. In practice, however, they may not be one-hot vectors strictly due to the following two reasons, which are widely existed in the supervised learning. First, it is hard to reach the global optima for training DNNs due to the strong non-convex property. Second, even a global optimum will be achieved on the training data, it is almost impossible to achieve a global optimum for all data (including the unobservable data). To manage this issue, patterns are clustered by locating the largest response of indicator features, i.e.,

$$c_i := \arg \max_h (I_{ih}), \quad h = 1, \cdots, k,$$

(15)

where $c_i$ is the clustering label of pattern $x_i$. Note that labels with high likelihood are assigned to patterns.

In summary, the developed SECT algorithm is listed in Algorithm 1. During each iteration, the algorithm alternately selects similar and dissimilar pairwise patterns via the fixed DNNs and trains the DNNs based on the selected pairwise patterns. When all pairwise patterns are considered for training and the objective function in Eq. (12) can not be improved further, the algorithm converges. Conclusively, patterns are clustered by locating the largest response of the learned indicator features.

5 Experiments

In this section, the clustering performance of the DSEC model are evaluated on several popular datasets with three frequently used metrics. Further, numerous ablation experiments are also conducted to systematically and comprehensively analyze the developed DSEC model. Specifically, the core code of the DSEC model is released at https://github.com/vector-1127/DSEC.

2. Relies on Keras [56] with the Theano [57] backend.
5.1 Datasets

We perform experiments on twelve popular datasets, including images, texts, and speeches. The number of patterns, the number of clusters, and dimensions of patterns are orderly listed in Table 1. As described in [30], [43], the training and testing patterns of each dataset are jointly utilized in our experiments.

Specifically, the image datasets include MNIST [58], CIFAR-10 [59], CIFAR-100 [59], STL-10 [60] and ILSVRC2012-1K [61]. For the CIFAR-100 dataset, additionally, the 20 superclasses are considered in our experiments. We randomly choose 10 subjects from the ILSVRC2012-1K dataset [61] and resize these images to $96 \times 96 \times 3$ to construct the ImageNet-10 dataset for our experiments. To compare the clustering methods on more complex dataset, we also randomly select 15 kinds of dog images from ILSVRC2012-1K to establish the fine-grained ImageNet-Dog dataset. The text datasets include 20NEWS [62], REUTERS [63], and REUTERS-10k [63] (which is a subset of the 20000 texts sampled from REUTERS following [43]). Three speech datasets, i.e., AudioSet-20, AudioSet-Music, and AudioSet-Human, are randomly chosen from AudioSet [64] for comparison in our experiments. In these datasets, AudioSet-Music and AudioSet-Human are two fine-grained datasets sampled from the coarse classes “Music” and “Human sounds” in AudioSet [64], respectively. Specifically, the term frequency-inverse document frequency features\(^3\) (TF-IDF) and the Mel-frequency cepstral coefficients\(^4\) are employed to recode texts and speeches, respectively.

5.2 Evaluation Metrics

Three popular metrics are utilized to evaluate the performance of clustering methods, including Normalized Mutual Information (NMI) [65], Adjusted Rand Index (ARI) [66], and clustering Accuracy (ACC) [67]. Specifically, these metrics range in \([0, 1]\), and higher scores signify more accurate clustering results are achieved.

5.3 Compared Methods

Several existing clustering methods are employed to compare with our approach. Specifically, the traditional methods, including K-means [11] and SC [31] are adopted for comparison. For the representation-based clustering approaches, as described in [43], we employ some unsupervised learning methods, including AE [18], DAE [19] and CILR [48], to learn feature representations and use K-means [11] to cluster data as a post processing. To a comprehensive comparison, recent single-stage methods, including CATGAN [51], GMVÆ [52], DEC [43], JULE-SF [30], and JULE-RC [30] are used for comparison.

5.4 Experimental Settings

For the traditional clustering methods, i.e., K-means [11] and SC [31], following the previous work [43], we concatenate HOG feature [16] and a $8 \times 8$ color map as inputs when we experiment on STL-10, ImageNet-10 and ImageNet-Dog. For the remaining datasets and methods, the pixel intensities on the image datasets and the extracted feature representations on the text and audio datasets are employed as inputs, respectively.

In our experiments, DNNs with the constraint layer are devised to learn indicator features (the details of the devised DNNs can be found in the supplementary material, available online). Specifically, we set $C_{u\theta}$ (the cosine similarity) as a default selection of the clustering constraint in our experiments. Since the prior probability of pairwise patterns belonging to different clusters is higher than to the same clusters, we set $u = 0.99$ and $l = 0.8$ for selecting similar and dissimilar pairwise patterns, respectively. The learning rate $\eta = \frac{e^{l} - 1}{e^{l} - e^{u}} = 0.0019$, where $e$ signifies the number of iterations and is set to 50. In SECT, we gradually select $m = 1000$ patterns as a batch to select labeled pairwise patterns. The normalized Gaussian initialization strategy [68] is utilized to initialize the devised DNNs. The RMSProp optimizer [69] where the initial learning rate is set to 0.001 is utilized to train the DNNs. The batch size is 32 in the learning procedure. For a reasonable evaluation, we perform 10 random restarts for all experiments and the average results are employed to compare with the others methods. Following with the previous works [19], [48], in addition, the unsupervised data augmentation technique in [48] is used to combat over-fitting and to guide clustering in our experiments.

5.5 Clustering

5.5.1 Quantitative Results

In Table 2, we report the quantitative results of these clustering methods on the experimental datasets. For each dataset, note that DSEC dramatically outperforms the others methods with significant margins on all the three evaluation metrics. The results imply that DSEC achieves state-of-the-art performance on the clustering task.

Further analysis, several tendencies can be observed from Table 2. First, the performance of the representation-based clustering methods (e.g., AE [18], CILR [48]) are superior to the traditional methods (e.g., K-means [11], SC [31]) currently. It indicates that the clustering techniques have only a minor impact on performance, while the representation extracting is more crucial. This scenario also exists in the supervised tasks, e.g., the representation learning plays crucial roles in the classification task [36]. Second, benefited from the effective representations, impressive improvements are achieved learned by these unsupervised methods, but

\(^3\) https://scikit-learn.org/stable
\(^4\) https://librosa.github.io/librosa/feature.html
The Clustering Results of the Compared Methods on the Experimental Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>MNIST [58]</th>
<th>CIFAR-10 [59]</th>
<th>CIFAR-100 [59]</th>
<th>STL-10 [60]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NMI</td>
<td>ARI</td>
<td>ACC</td>
<td>NMI</td>
</tr>
<tr>
<td>K-means [11]</td>
<td>0.4997</td>
<td>0.3652</td>
<td>0.5723</td>
<td>0.0871</td>
<td>0.0487</td>
</tr>
<tr>
<td>SC [31]</td>
<td>0.6626</td>
<td>0.5214</td>
<td>0.6958</td>
<td>0.1028</td>
<td>0.0853</td>
</tr>
<tr>
<td>AE [18]</td>
<td>0.7257</td>
<td>0.6139</td>
<td>0.8123</td>
<td>0.2393</td>
<td>0.1689</td>
</tr>
<tr>
<td>DAE [19]</td>
<td>0.7563</td>
<td>0.6467</td>
<td>0.8316</td>
<td>0.2506</td>
<td>0.1627</td>
</tr>
<tr>
<td>CILR [48]</td>
<td>0.7986</td>
<td>0.7756</td>
<td>0.8855</td>
<td>0.1105</td>
<td>0.0612</td>
</tr>
<tr>
<td>GMVAE [52]</td>
<td>0.7964</td>
<td>0.7670</td>
<td>0.8854</td>
<td>0.2847</td>
<td>0.1873</td>
</tr>
<tr>
<td>CatGAN [51]</td>
<td>0.7637</td>
<td>0.7360</td>
<td>0.8279</td>
<td>0.2646</td>
<td>0.1757</td>
</tr>
<tr>
<td>JULE-SF [30]</td>
<td>0.9063</td>
<td>0.9139</td>
<td>0.9592</td>
<td>0.1919</td>
<td>0.1357</td>
</tr>
<tr>
<td>JULE-RC [30]</td>
<td>0.9130</td>
<td>0.9270</td>
<td>0.9640</td>
<td>0.1923</td>
<td>0.1377</td>
</tr>
<tr>
<td>DEC [43]</td>
<td>0.7716</td>
<td>0.7414</td>
<td>0.8430</td>
<td>0.2568</td>
<td>0.1607</td>
</tr>
<tr>
<td>DSEC</td>
<td>0.9522</td>
<td>0.9631</td>
<td>0.9833</td>
<td>0.4379</td>
<td>0.3399</td>
</tr>
</tbody>
</table>

The best results are highlighted in **bold**.

...they are limited, especially compared against the single-stage clustering methods (e.g., JULE [30], DSEC). This demonstrates that the single-stage clustering methods can observably improve the clustering performance, since more excellent representations will be captured during clustering. Thirdly, dramatical superiorities are yielded by DSEC on the CIFAR-100, ImageNet, and AudioSet datasets, which signifies that DSEC can be utilized on large-scale datasets, not merely limited to some simple datasets (e.g., MNIST). Fourthly, competitive results are obtained by DSEC on the image, text and audio datasets concurrently, which demonstrates that DSEC is efficient for managing general clustering tasks in practice. Finally, the performance of JULE approximates to our method on MNIST. However, there is a conspicuous margin on the other datasets. A possible reason is that the initialization strategy of JULE is invalidated on intricate datasets, which degenerates the performance consequently. In contrast, DSEC alleviates the dependence on additional techniques by introducing the Self-Evolution Clustering Training algorithm.

5.5.2 Results Visualization

The clustering results of DSEC on our experimental datasets are intuitively illustrated in Fig. 4. For clarity, the learned indicator features are mapped in the 2-dimensional space with t-SNE [70]. In the figure, different colors correspond to different clusters. Visually, this illustration demonstrates that the developed DSEC model can gradually cluster patterns with the same indicator features if patterns belong to the same clusters, and cluster patterns with different indicator features otherwise.

5.5.3 Indicator Features Visualization

In Fig. 5, we qualitatively analyze the indicator features learned by DSEC on the experimental datasets. For clarity,
Fig. 4. The clustering results of DSEC on the experimental datasets. For each dataset, the names of the datasets are written on the upward side, the clustering results in the initial, intermediate and final stages are orderly illustrated in the top, intermediate and bottom lines. In addition, the clustering accuracies at the different stages are listed under the results. Large figures can be found in the supplement, available online.

Fig. 5. The indicator features of the first ten clusters on MNIST, ImageNet-10, 20NEWS and AudioSet-20. For each dataset, ground-truth labels are written on the upward side, indicator features are shown on the right side of examples, followed by typical failures at the bottom.
and spectrograms are employed to visualize text and audio patterns, respectively. We observe that the same neurons will be distinctly activated in the indicator features if the patterns belong to the same clusters, and the failure cases always possess different fine-grained information compared with the correct cases. That is, DSEC attempts to learn high-level representations, rather than a simple combination of lower-level representations. This is the reason why more complicated images, such as the airliner and airship images in ImageNet-10, can be distinguished by DSEC. Interestingly, most of the indicator features of the failure modes appear reasonable. For example, in terms of car, only the other types of vehicle (e.g., truck) might be considered as plausible labels, rather than the clusters beyond vehicle. It implies that more interpretable and high-level knowledge is captured by DSEC for clustering.

5.5.4 The Promotion of DSEC

To observe the self-evolution capability of SECT, Fig. 6 depicts the precisions of the selected similar and dissimilar patterns during the learning stages on the experimental datasets. It shows that SECT can gradually select more accurate pairwise patterns for training, which verifies that SECT is steadily self-evolution as the learning progresses. Furthermore, two tendencies can be observed from the illustration. First, the precisions of the dissimilar patterns always high than the dissimilar patterns, which is in agreement with the analysis in Theorem 2. Second, the promotions are mainly occurred during initial stages, which is similar to the tendency in the supervised learning [71]. It also indicates that the initial status are significant for DSEC, which is further verified with an experiment in Section 5.6.8.

5.6 Ablation Study

In this section, extensive ablation studies are performed to synthetically analyze the DSEC model. Specifically, the experimental settings always inherit from the statements in Section 5.4, except some special settings in each study.

5.6.1 Effect of SECT

In order to empirically demonstrate the practicality of the SECT algorithm, DSEC-SECT that utilizes all pairwise patterns to train the networks is conducted to compare with DSEC. In DSEC-SECT, specifically, we set $u = l = 0.99$ for the beginning, followed by an annealing phase which decreases linearly to $u = l = 0.8$. Table 3 shows that DSEC achieves better performance than DSEC-SECT. Further analysis, since DNNs are initialized randomly, more noisy data will be considered during training in DSEC-SECT. Contrary to DSEC-SECT, DSEC can select highly confident pairwise patterns based on the SECT algorithm. By using these selected pairwise patterns, DSEC can begin with more refined clusters and improve the clustering performance finally.

5.6.2 Effect of Label Inference

To evaluate the effect of the label inference tactic in Eq. (15), we employ the traditional methods, including K-means [11] and SC [31] to cluster patterns via the generated indicator features. From the results listed in Table 3, note that the label inference tactic outperforms the traditional methods. Furthermore, compared with these traditional methods, the label inference tactic is straightforward and simple since DSEC just needs to locate the largest response of the generated indicator features to yield clustering labels.

5.6.3 Impact of Number of Clusters

We conduct an experiment on 9 different datasets to study the stabilities of these methods by varying the number of clusters. For each dataset, we randomly select $k$ subject from ILSVRC2012-1K [61]. By varying $k$ between 10 and 50 with an interval 5, 9 different datasets are established. As the number of clusters increases, from Fig. 7a, all the methods are generally degraded, which is because more uncertainty may be introduced. However, contrary to other methods, the superiority of DSEC is still kept, which implies that DSEC has outstanding capability to deal with clustering tasks with larger number of clusters.

5.6.4 Impact of Number of Patterns

To observe the effect of the number of patterns to the clustering methods, we vary it between 10000 and 60000 with an interval 10000 on CIFAR-10. Fig. 7b visually illustrates that the performance of most methods improves with more patterns, which implies that more patterns are beneficial for clustering. Furthermore, we observe that the performance of DSEC increases rapidly when more patterns are considered, which is similar to the influence of the number of labeled patterns in the supervised learning tasks. A significant reason is that sufficient patterns are essential to learn numerous parameters in deep models to map patterns from inputs to outputs.

5.6.5 Performance on Imbalanced Datasets

To study the performance of DSEC on imbalanced datasets, an additional experiment is executed on MNIST. In the experiment, we randomly sample five subsets from MNIST with various minimum retention rates. For the minimum retention rate $r$, the patterns of the first class will be kept with probability $r$ and the last class with probability 1, with the other classes linearly in between. From Fig. 7c, we observe

5.6.1 Effect of SECT

In order to empirically demonstrate the practicality of the SECT algorithm, DSEC-SECT that utilizes all pairwise

patterns to train the networks is conducted to compare with DSEC. In DSEC-SECT, specifically, we set $u = l = 0.99$ for the beginning, followed by an annealing phase which decreases linearly to $u = l = 0.8$. Table 3 shows that DSEC achieves better performance than DSEC-SECT. Further analysis, since DNNs are initialized randomly, more noisy data will be considered during training in DSEC-SECT. Contrary to DSEC-SECT, DSEC can select highly confident pairwise patterns based on the SECT algorithm. By using these selected pairwise patterns, DSEC can begin with more refined clusters and improve the clustering performance finally.

5.6.2 Effect of Label Inference

To evaluate the effect of the label inference tactic in Eq. (15), we employ the traditional methods, including K-means [11] (DSEC+K-means) and SC [31] (DSEC+SC) to cluster patterns via the generated indicator features. From the results listed in Table 3, note that the label inference tactic outperforms the traditional methods. Furthermore, compared with these traditional methods, the label inference tactic is straightforward and simple since DSEC just needs to locate the largest response of the generated indicator features to yield clustering labels.

5.6.3 Impact of Number of Clusters

We conduct an experiment on 9 different datasets to study the stabilities of these methods by varying the number of clusters. For each dataset, we randomly select $k$ subject from ILSVRC2012-1K [61]. By varying $k$ between 10 and 50 with an interval 5, 9 different datasets are established. As the number of clusters increases, from Fig. 7a, all the methods are generally degraded, which is because more uncertainty may be introduced. However, contrary to other methods, the superiority of DSEC is still kept, which implies that DSEC has outstanding capability to deal with clustering tasks with larger number of clusters.

5.6.4 Impact of Number of Patterns

To observe the effect of the number of patterns to the clustering methods, we vary it between 10000 and 60000 with an interval 10000 on CIFAR-10. Fig. 7b visually illustrates that the performance of most methods improves with more patterns, which implies that more patterns are beneficial for clustering. Furthermore, we observe that the performance of DSEC increases rapidly when more patterns are considered, which is similar to the influence of the number of labeled patterns in the supervised learning tasks. A significant reason is that sufficient patterns are essential to learn numerous parameters in deep models to map patterns from inputs to outputs.

5.6.5 Performance on Imbalanced Datasets

To study the performance of DSEC on imbalanced datasets, an additional experiment is executed on MNIST. In the experiment, we randomly sample five subsets from MNIST with various minimum retention rates. For the minimum retention rate $r$, the patterns of the first class will be kept with probability $r$ and the last class with probability 1, with the other classes linearly in between. From Fig. 7c, we observe
<table>
<thead>
<tr>
<th>Dataset</th>
<th>MNIST [58]</th>
<th>CIFAR-10 [59]</th>
<th>CIFAR-100 [59]</th>
<th>STL-10 [60]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metric</td>
<td>NMI</td>
<td>ARI</td>
<td>ACC</td>
<td>NMI</td>
</tr>
<tr>
<td>DSEC-SECT</td>
<td>0.9271</td>
<td>0.9310</td>
<td>0.9682</td>
<td>0.4034</td>
</tr>
<tr>
<td>DSEC + K-means</td>
<td>0.9504</td>
<td>0.9618</td>
<td>0.9826</td>
<td>0.4313</td>
</tr>
<tr>
<td>DSEC + SC</td>
<td>0.9538</td>
<td>0.9623</td>
<td>0.9981</td>
<td>0.4209</td>
</tr>
<tr>
<td>DSEC</td>
<td>0.9522</td>
<td>0.9631</td>
<td>0.9833</td>
<td>0.4379</td>
</tr>
</tbody>
</table>

Table 3: The Comparison of the Clustering Results on the Experimental Datasets with Various Scenarios

The best results are highlighted in bold.

5.6.6 Impact of Feature Extraction

In order to explore the impact of feature extraction, we employ two different approaches on 20NEWS, i.e., the bag of word (BoW) and TF-IDF features, to represent the texts in 20NEWS. As shown in Fig. 7d, feature extraction can significantly influence all these clustering methods, which is akin to the impact in supervised learning, i.e., better features bring better performance. Furthermore, it is worth noting that the superior performance with significant margins is consistently achieved by DSEC, which verifies the stability of DSEC.

5.6.7 Contribution of Clustering Constraint

To investigate the contribution of clustering constraints in Eq. (9), Fig. 8a exhibits the distributions of the elements in the generated indicator features on MNIST and ImageNet-10 in the initial and final stages. In this experiment, we severally count the number of the elements in the following ten disjoint intervals, i.e., $[0, 0.1), [0.1, 0.2), \ldots, [0.9, 1]$. Initially, the major elements of the indicator features locate in $[0, 0.1)$ and $[0.1, 0.2)$. Finally, most elements move to $[0, 0.1)$ and $[0.9, 1]$. It implies that the generated indicator features are sparse and the non-zero elements tend to be 1 simultaneously. This evolution also corresponds to our target that DSEC attempts to learn one-hot vectors to encode and cluster patterns.

5.6.8 Contribution of Pre-Training

To explore the contribution of pre-training on the DSEC model, we perform DSEC on the networks pre-trained by the unsupervised and supervised pre-training techniques. As an unsupervised frequently used method, the AE [18] (DSEC + AE) is employed as a basic method to pre-train the networks in a supervised way. For the supervised pre-training, DSEC+AE(n) represents that n labeled patterns are used to pre-train the networks in a supervised manner. From Fig. 8b, the pre-training techniques can significantly improve the performance of DSEC. It is expected since the pre-training can initialize the networks to some justified status, which may assist DSEC to select more accurate labeled pairwise patterns for clustering during initial stages. Further, the supervised pre-training is always superior to the unsupervised pre-training. A considerable reason is that labeled patterns

7. https://pypi.python.org/pypi/bagofwords

Authorized licensed use limited to: University of Chinese Academy of Sciences.CAS. Downloaded on March 16,2020 at 02:45:29 UTC from IEEE Xplore. Restrictions apply.
can guide the networks to more informative initial status to achieve the better performance in DSEC.

5.6.9 Influence of Network Configuration
Three popular variants of VGG [34], i.e., VGG-13, VGG-16 and VGG-19, are employed to study the influence of network configuration on DSEC. Fig. 8c visually shows the difference between these networks. A possible reason is that the overfitting causes the performance difference in Fig. 10 (Fig. 8c in the revised manuscript). Specifically, VGG-19 possesses more 6 million parameters and 10 million parameters than VGG-16 and VGG-13, respectively. Since there are only 13000 images in ImageNet-10, VGG-19 is easy to over-fitting and degenerates the performance consequently. Although the difference is yielded, however, the clustering accuracies of all three networks (VGG-13, VGG-16 and VGG-19) are approximately equal to 0.60, which outperforms the compared methods with significant margins (+0.20). These results indicate that DSEC can achieve superior performances on diverse networks.

5.6.10 Sensitivity of Initialized Thresholds
To investigate the stability of the SECT algorithm, we compare the performance of DSEC under various initial values of $l$ and $u$. From Fig. 9a, we observe that the initial values have a significantly influence on the performance of DSEC. It is because DNNs are initialized randomly, more noisy labeled pairwise patterns may be employed for training if $l$ is too large or $u$ is too small. Further analysis, more appropriate values can assist DSEC to select highly confident and enough labeled pairwise patterns. By learning with these selected pairwise patterns, DSEC can begin with more refined clusters and achieve the better clustering results finally.

5.6.11 Sensitivity of Clustering Constraint
A series of experiences are conducted on ImageNet-Dog to study the sensitivity of clustering constraints on DSEC. As depicted in Fig. 9b, the clustering constraint $C$ is incapable of yielding one-hot vectors to recode patterns, even if it is possible in theory. A conceivable reason is that the search space of the clustering constraint $C$ is too large and the backpropagation learning algorithm cannot attain the theoretical values. Compared with $C$, the search space of $C_p$ is significantly reduced to efficiently guide DSEC to learn one-hot vectors. Furthermore, the impact of $C_p$ varies with the hyperparameter $p$. Specifically, the peak performance of DSEC is obtained when $p$ is approximately in the range $[1, 2]$ and the too large (e.g., 4) or too small (e.g., 0.5) parameter $p$ may degrade the learning stability and the convergence speed, which are predictable. Intrinsically, the optimal indicator features purely consist of two essential elements, i.e., 0 and 1. For too small parameters $p$, $C_p$ attempts to learn small number to compose the optimal indicator features, which is hard to learn the number 1. Contrary to too small $p$, too large $p$ is inefficient to learn the number 0 to form the optimal indicator features as well.

To verify the necessity of the clustering constraints, we compare our model with two baselines, i.e., Null1 and Null2. Specifically, Null1 means that the clustering constraint is omitted and our label inference is used for clustering, and Null2 indicates that the clustering constraint is omitted and K-means is used for clustering. The results in Fig. 9b show that our clustering constraints are necessary for our model. If such constraints are omitted, Theorem 1 is not satisfied. As a result, DSEC can learn feature representations only, and an additional clustering method is required to cluster data.

5.6.12 Benefit for Semi-Supervised Learning Tasks
A semi-supervised task, i.e., the classification task with numerous unlabeled patterns and a small number of labeled patterns, is executed on the ImageNet-10 dataset to study the availability of the networks pre-trained by DSEC. In Fig. 9c, $L(n)$ signifies that only $n$ labeled patterns are employed to
train networks from randomly initialized status, and $L(n)$ indicates that $n$ labeled patterns are utilized to fine-tune the networks pre-trained by DSEC. From the figure, intuitively, overall the experimental results empirically confirm that the superior classification results can be achieved by the pre-trained networks with stable training processes. It demonstrates that the networks can be trained to some justified status and can be modified based on very limited labeled patterns.

5.6.13 Filters Visualization

To study whether the proposed DSEC model can learn meaningful parameters, a DNN (the configurations can be found in the supplementary material, available online) is devised to present the learned filters in the first convolutional layer. As shown in Fig. 10, DSEC attempts to learn a variety of frequency and orientation selective filters, as well as various colored blocks, which are similar to the filters learned with label information. This verifies that DSEC is efficient of learning informative filters to capture high-level representations in a purely unsupervised manner.

6 Conclusion

In this paper, we have developed a DNN-based clustering method, i.e., deep self-evolution clustering, which recast the clustering task into a DNN to judge whether pairwise patterns belong to the same clusters. To generate more informative representations for clustering, constrained indicator features have been introduced into the DNN to recode patterns with one-hot vectors. Since the ground-truth similarities are unavailable in clustering, the self-evolution clustering training algorithm has been proposed to select similar and dissimilar pairwise patterns and to train the DNN via the selected pairwise patterns alternately. Conclusively, DSEC cluster data by locating the largest response of the generated indicator features purely. In comparison with existing approaches, our approach has achieved superior performance on image, text, and audio datasets concurrently.

Future work may include exploring more justified clustering constraints to learn the indicator features and generalizing DSEC to select the number of clusters automatically. For the first problem, a series of clustering constraints have been introduced as some pioneering work. Nevertheless, which one is the best is an open problem. It will be interesting in the future to adaptively learn clustering constraints to improve the clustering performance. For the second problem, we have predefined the number of clusters. However, it is hard to be defined before clustering. Inspired by the DCD method [14], a possible direction is to introduce a residual similar to Normalized Output Similarity Approximation Clusterability [14] into our model. Consequently, the number of clusters may be optimized and selected automatically.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 91646207, 61773377, and 61573352, and the Beijing Natural Science Foundation under Grant L172053. We would like to thank Lele Yu, Jie Gu, Cheng Da, and Tingzhao Yu for their invaluable contributions in shaping the early stage of this work.

References

Jianlong Chang

received the BS degree from the School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, China, in 2015. He is currently working toward the PhD degree in the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China.

His research interests include pattern recognition, machine learning and relation-based deep learning.
Gaofeng Meng received the BS degree in applied mathematics from Northwestern Polytechnical University, in 2002, and the MS degree in applied mathematics from Tianjing University, in 2005, and the PhD degree in control science and engineering from Xi’an Jiaotong University, in 2009. He is currently an associate professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. He also serves as an associate editor of Neurocomputing. His current research interests include document image processing, computer vision, and pattern recognition. He is a senior member of the IEEE.

Lingfeng Wang received the BS degree in computer science from Wuhan University, Wuhan, China, in 2007, and the PhD degree from the Institute of Automation, Chinese Academy of Sciences, in 2013. He is currently an associate professor with the National Laboratory of Pattern Recognition of Institute of Automation, Chinese Academy of Sciences. His research interests include computer vision and image processing. He is a member of the IEEE.

Shiming Xiang received the BS degree in mathematics from Chongqing Normal University, Chongqing, China, in 1993, the MS degree from Chongqing University, Chongqing, China, in 1996, and the PhD degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2004. From 1996 to 2001, he was a Lecturer with the Huazhong University of Science and Technology, Wuhan, China. He was a postdoctorate candidate with the Department of Automation, Tsinghua University, Beijing, China, until 2006. He is currently a professor with the National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include pattern recognition, machine learning, and computer vision. He is a member of the IEEE.

Chunhong Pan received the BS degree in automatic control from Tsinghua University, Beijing, China, in 1987, the MS degree from Shanghai Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, China, in 1990, and the PhD degree in pattern recognition and intelligent system from Institute of Automation, Chinese Academy of Sciences, Beijing, in 2000. He is currently a professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include computer vision, image processing, computer graphics, and remote sensing. He is a member of the IEEE.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.