Discerning Feature Supported Encoder for Image Representation

Shuyang Wang, Zhengming Ding, *Member, IEEE*, and Yun Fu, *Fellow, IEEE*

***Abstract*— Inspired by the recent successes of deep architecture, the auto-encoder and its variants have been intensively explored on image clustering and classification tasks by learning effective feature representations. Conventional auto-encoder attempts to uncover the data's intrinsic structure, by constraining the output to be as much identical to the input as possible, which denotes that the hidden representation could faithfully reconstruct the input data. One issue that arises, however, is that such representations might not be optimized for specific tasks, e.g., image classification and clustering, since it compresses not only the discriminative information but also a lot of redundant or even noise within data. In other words, not all hidden units would benefit the specific tasks, while partial units are mainly used to represent the task-irrelevant patterns. In this paper, a general framework named *discerning feature supported encoder* (DFSE) is proposed, which integrates the auto-encoder and feature selection together into a unified model. Specifically, the feature selection is adapted to learned hidden-layer features to capture the task-relevant ones from the task-irrelevant ones. Meanwhile, the selected hidden units could in turn encode more discriminability only on the selected task-relevant units. To this end, our proposed algorithm can generate more effective image representation by distinguishing the task-relevant features from the task-irrelevant ones. Two scenarios of the experiments on image classification and clustering are conducted to evaluate our algorithm. The experiments on several benchmarks demonstrate that our method can achieve better performance over the state-of-the-art approaches in two scenarios.**

***Index Terms*— Image representation, feature extraction, learning systems.**

I. Introduction

THE real-world image data are commonly in very high dimensions, which leads to the considerable escalation of the computational time and space. Even though a number of studies in the literature have examined strategies to deal with high-dimensional data, it is still a crucial difficulty associated with many practical image learning problems caused by the curse of dimensionality [1]. On the other hand, dealing with a large number of features will often deteriorate the performance due to the large noise and irrelevant sensory patterns contained in the image data. Practically, not all features are equally discriminative and important to image classification or clustering tasks, as most of them are often highly correlated to each other or even redundant [2]. Consequently, it is necessary to learn a low-dimensional image representation to preserve its inherent information.

Dimensionality reduction is the common approach to deal with the curse of dimensionality, which can be categorized into two fashions: (1) feature selection [2], [3], which selects a subset of most representative or discriminative features from the input feature set and (2) feature extraction (or subspace learning) [4], [5], which transforms the original input features to a lower dimensional subspace. A lot of research activities have exploited on jointly feature selection and feature extraction to benefit from each other. Zou *et al.* [6] proposed sparse subspace learning methods and attempted to solve this problem based on l_2-norm and l_1-norm regularization. Gu *et al.* [7] developed a joint framework to integrate subspace learning and feature selection via $l_{2,1}$-norm to achieve a row-sparse linear projection. These methods all adopt sparse constraint to implement the feature selection and they all employ shallow structures to seek linear projection. Therefore, these methods cannot uncover the rich hierarchical information within the complex data.

As a method of nonlinear dimensionality reduction, the auto-encoder and its variants have attracted increasing attention recently [8]–[10], and achieve better results on a series of tasks [11], [12]. The conventional auto-encoder train the network to seek a compressed approximation by forcing the output to be as much close to the target as possible. However, one problem will be brought up by this scheme that the major part of the high-level feature units could blindly contribute to represent the patterns unrelated with later classification or clustering tasks. Although the effort to incorporate supervision [13] has been deployed in recent literature, there is still an absence of works which separately consider high-level features relevant or irrelevant in term of the task. It is inappropriate to endow the discriminability to those hidden units which primarily used to reconstruct the distracting or noisy patterns of the input.

To address this issue, we propose to integrate auto-encoder and feature selection into a unified model (Fig.1) to seek effective image representation. Intuitively, the feature selection is adapted to learned high-level features to capture

Manuscript received July 12, 2017; revised August 14, 2018 and January 10, 2019; accepted February 1, 2019. Date of publication February 21, 2019; date of current version June 13, 2019. This was supported in part by the NSF IIS under Award 1651902 and in part by the U.S. Army Research Office under Award W911NF-17-1-0367. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Weisi Lin. *(Corresponding author: Shuyang Wang.)*

S. Wang is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 USA (e-mail: shuyangwang@ece.neu.edu).

Z. Ding is with the Department of Computer, Information and Technology, Indiana University–Purdue University Indianapolis, Indianapolis, IN 46202 USA (e-mail: zd2@iu.edu).

Y. Fu is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 USA, and also with the Khoury College of Computer and Information Sciences, Northeastern University, Boston, MA 02115 USA (e-mail: yunfu@ece.neu.edu).

Digital Object Identifier 10.1109/TIP.2019.2900646

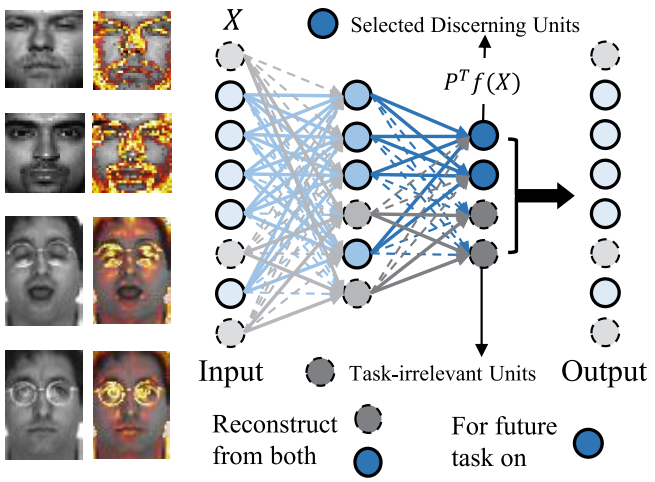


Fig. 1. The framework of our proposed algorithm. The hidden layer is learned with the constraint of feature selection criterion to separate task-irrelevant units from discerning ones, while those selected units are encoded to compress more important patterns. All of the units contribute to reconstruct the input, while only selected ones are used for later tasks. The left column images are the input sample from YaleB and ARface datasets, while the right column images indicate the region where its reconstruction involves more of learned discerning feature.

the discriminative ones from the task-irrelevant units. In the meanwhile, those selected high-level features are enforced to be trained to compress more discriminate patterns. Therefore, the proposed framework not only functions dynamic feature selection on learned hidden units (i.e., high-level features), but also compress the important information mainly on task-relevant hidden units, while ignoring the irrelevant patterns or noise into deserted units. We emphasize our major contributions in three folds as follows:

- We propose Discerning Features Supported Encoder (DFSE) for image representation, which selects the discerning high-level features and, furthermore, enhances the discriminative ability on the selected units.
- The framework can be easily extended to both classification or clustering scenarios, by simply shifting the feature selection criterion (Fisher score and Laplacian score in this paper) on the hidden layer. Both supervised and unsupervised version are demonstrated.
- The proposed DFSE can be easily adopt to form a stacked deep network. We evaluate the proposed DSFE method through both classification and clustering schemes, as well as visualization tasks to intuitively demonstrate the effectiveness of our proposed framework.

The rest of the paper is organized as follows. We first briefly introduce the related works in Section II. Our new proposed algorithm along with its optimization solution, differences with other methods and stacked version are introduced in Section III. Section IV shows the experimental results and analysis, following with our conclusions in Section V.

II. Related Work

In this section, we introduce the related works from two lines, one is feature selection and the other is auto-encoder. Generally, both belong to feature learning. Feature selection aims to select features in the original space, while auto-encoder targets at transforming the original features to a new hidden space.

Feature selection approaches can be roughly classified into filters, wrappers, and embedded methods. In the filter based methods, a subset of features are selected through ranking the features based on some well-defined performance criterion. The poorly informative features tend to be filtered out prior to the learning algorithm. In other words, the filter methods are classifier-independent and only based on the intrinsic structures of the data. Wrapper approaches adopt the classification performance itself as the evaluation criterion to search for a good subset of features. Compared to filter methods, wrappers are classifier-specific and the feature selection is wrapped in the learning algorithm that will ultimately be employed. Embedded methods perform feature selection as part of the learning procedure. In contrast with filter methods, wrapper and embedded methods usually achieve better results because they are tightly related to the in-built specific classifiers. However, it in turn restrains their generality and causes more computational cost. In this paper, we focus on incorporating the filter-based methods to the auto-encoder framework.

The past decade has witnessed a number of criteria proposed for filter based feature selection, such as Fisher score [14], Relief [15], Laplacian score [16], Hilbert Schmidt Independence Criterion (HSIC) [17], Information Gain and Trace Ratio criterion [18]. It is often prohibitively expensive in computational cost to search a subset of feature combination from original space to maximize the criterion. Accordingly, the conventional feature-level selection algorithms, instead of selecting at subset-level, calculate the score of each feature independently at first then select the top-ranking features. Those features selected one by one are yet suboptimal, since the subset-level score are neglected, which leads to either discard good combination of features or preserve redundant features. To address this issue, a globally optimal solution were proposed by Gu *et al.* [14] and Nie *et al.* [18] based on Trace Ratio criterion and Fisher score respectively.

Auto-encoder (AE) was proposed as an efficient technique for dimensionality reduction and deep structures pre-training [8], which has drawn increasing attentions in computer vision fields in recent years. Variations of AE variants have been proposed most recently to handle different learning tasks, e.g., domain generalization [19], multi-view learning [21] and transfer learning [20]. Yu *et al.* [10] proposed an embedding with auto-encoder regularization (EAER) framework, which utilize graph embedding structure to guide the auto-encoder reconstruction. However, learning is still challenging when the data contains lots of irrelevant patterns. There is still lack of auto-encoder based method that considers two parts of the hidden unites separably, which are task-relevant ones and -irrelevant ones. In our proposed framework, feature selection is applied on the hidden units of auto-encoder in order to guide the encoder to compress task-relevant and -irrelevant patterns into two groups of high-level features.

This paper is the extension of our previous conference work [22]. Our previous work proposed an auto-encoder guided with feature selection in a supervised fashion, where we

explored a Fisher score to select the hidden units under the label information. Thus, our model can only work for the classification problem. In this extension, we naturally extend our model to make it more general to handle both supervised learning and unsupervised learning. We can easily shift the feature selection criterion according to different scenarios. More specifically, we can adopt Fisher score as our supervised feature selection method for classification, while Laplacian score as the unsupervised feature selection method for clustering. In this way, our model can deal with more real-world problems based on the availability of label information. To this end, we evaluate on more benchmarks in both classification and clustering tasks, which demonstrate our model works in both scenarios compared with other algorithms.

III. The Proposed Algorithm

We will first give the preliminary in this section and followed by emphasizing our motivation. The detailed model introduction and optimization are provided by jointly selecting features and training auto-encoder, including supervised version and unsupervised one. Then, the discussion of two most relevant algorithms will be given as well.

A. Preliminary and Motivations

Assume $X \in \mathbb{R}^{d\times n}$ is the training data with n samples and $x_i \in \mathbb{R}^d$ represents the i-th sample. A single layer auto-encoder [8] usually contains two parts, which are encoder and decoder. The input x_i will be first map to a hidden representations by the encoder, denoted as f, and then reconstructed back to the itself with the decoder, denoted as g. A typical objective function with square loss can be formulated as:

$$\min_{W_1,b_1,W_2,b_2} \sum_{i=1}^{n} \left\| x_i - g(f(x_i)) \right\|_2^2, \tag{1}$$

where $\{W_1 \in \mathbb{R}^{r\times d}, b_1 \in \mathbb{R}^r\}, \{W_2 \in \mathbb{R}^{d\times r}, b_2 \in \mathbb{R}^d\}$ are learned parameter sets include weighted matrices and offset vectors. Specifically, we have $f(x_i) = \varphi(W_1 x_i + b_1)$ and $g(f(x_i)) = \varphi(W_2 f(x_i) + b_2)$, where $\varphi(\cdot)$ is an non-linear activation function, such as the tanh function or the sigmoid.

As mentioned previously, data reconstruction requires all hidden units to be conducive to capturing the intrinsic information of input data. Actually, not all hidden-layer features are equally important for our classification task. For instance, those units play a key role to compress the background information in an object image should help a little in later object recognition task. These units are considered as task-irrelevant ones in our framework, which are unwanted in the final feature set.

Thus, two conclusions could be drawn from the aforementioned discussion: 1) it is inappropriate and counterproductive to endow the entire hidden units with discriminative ability, the discerning information should be only encoded into the selected task-relevant features; and 2) feature selection could be conducive to distinguishing discerning units out of task-irrelevant units. Based on above assumption, we propose our discerning feature selection supported auto-encoder model through a joint learning scheme.

B. Discerning Feature Supported Encoder

In this section, the proposed Discerning Feature Supported Encoder (DFSE) is introduced by integrating feature selection on the hidden layer of auto-encoder as a joint learning framework. Suppose training data $X \in \mathbb{R}^{d\times n}$ has visual descriptor dimensionality d and number of data samples n. Our proposed framework is formulated as:

$$\min_{W_1,W_2,b_1,b_2,P} \frac{1}{2}\|X - g(f(X))\|_F^2 + \frac{\lambda}{2}\mathcal{C}(P, f(X)) \tag{2}$$

where $f(X) = \sigma(W_1 X + B_1)$, $g(f(X)) = \sigma(W_2 f(X) + B_2)$, B_1, B_2 are the n-repeated column copy of b_1, b_2, respectively. $\mathcal{C}(P, f(X))$ is the feature selecting criterion with a selection matrix P performs as the regularization term on hidden units $f(X)$. In detail, i-th column vector in P denoted by $p_i \in \mathbb{R}^z$ has the form,

$$p_i = [\underbrace{0, \cdots, 0}_{j-1}, 1, \underbrace{0, \ldots, 0}_{z-j}]^\top. \tag{3}$$

where z denotes the number of hidden units and j represents that the j-th units is selected into the subset of new features $y \in \mathbb{R}^m$ by this column vector p_i. Therefore the procedure of feature selection can be formalize as finding a selection matrix P to select a feature subset $Y = P^\top f(X)$ from original feature $f(X)$, so that the appropriate criterion $\mathcal{C}(P, f(X))$ is optimized.

Regularly, depends on the usage of label information in the training procedure, feature selection can be roughly split into unsupervised models, e.g., Laplacian score [16], and supervised models, e.g., Fisher score [14]. In order to deal with different scenarios in real world, we formulate the feature selection criterion $\mathcal{C}(P, f(X))$ into the following regularizer as:

$$\mathcal{C}(P, f(X)) = \frac{\mathrm{tr}(P^\top f(X) L_w f^\top(X) P)}{\mathrm{tr}(P^\top f(X) L_b f^\top(X) P)}, \tag{4}$$

which provides a general graph framework for feature selection. Different ways of constructing the weight matrices L_w and L_b will bring about different unsupervised, semi-supervised or supervised feature selection models. We will discuss more in the later section. The task of feature selection is to seek the feature selection with the minimum score by solving the following optimization problem:

$$P = \arg\min_{P} \frac{\mathrm{tr}(P^\top f(X) L_w f^\top(X) P)}{\mathrm{tr}(P^\top f(X) L_b f^\top(X) P)}, \tag{5}$$

Unfortunately, due to the nonexistent of closed-form solution, a straightforward issue to optimize this equation is not available. Therefore, instead of dealing with the above trace-ratio problem directly, many works trying to achieve a globally optimal solution by transforming it to an equivalent trace-difference problem [18], [23].

From Eq.(4), we have the global minimum γ^* of the subset-level criterion score $\mathcal{C}(P, f(X))$ satisfying,

$$\gamma^* = \arg\min \frac{\mathrm{tr}(P^\top f(X) L_w f^\top(X) P)}{\mathrm{tr}(P^\top f(X) L_b f^\top(X) P)},$$

that is to say,

$$\begin{aligned}&\frac{\mathrm{tr}(P^\top f(X)L_w f^\top(X)P)}{\mathrm{tr}(P^\top f(X)L_b f^\top(X)P)} \geq \gamma^*, \quad \forall\Phi(P)\\ &\Rightarrow \mathrm{tr}(P^\top f(X)(L_w-\gamma^* L_b)f^\top(X)P)\geq 0, \quad \forall\Phi(P)\\ &\Rightarrow \min_{\Phi(P)} \mathrm{tr}(P^\top f(X)(L_w-\gamma^* L_b)f^\top(X)P)=0. \end{aligned} \tag{6}$$

Then, we can derive the function of the global optimal γ while considering others as constant so that:

$$r(\gamma)=\arg\min_{\Phi(P)} \mathrm{tr}(P^\top f(X)(L_w-\gamma L_b)f^\top(X)P). \tag{7}$$

Note that the above function is monotonically increasing [18]. Accordingly, we can convert the finding of global optimal γ to a trace-difference problem by solving Eq.(7). After the above trace-difference regularization is applied on the hidden layer to constrain the auto-encoder optimization, our final objective function is reformulated as:

$$\begin{aligned}\min_{W_1,W_2,b_1,b_2,P} \mathcal{L}=&\frac{1}{2}\|X-g(f(X))\|_F^2\\ &+\frac{\lambda}{2}\mathrm{tr}(P^\top f(X)(L_w-\gamma L_b)f^\top(X)P),\end{aligned} \tag{8}$$

where the parameter λ works as a balance between auto-encoder and feature selection regularizer. $\gamma = \frac{\mathrm{tr}(P^\top f(X)L_w f^\top(X)P)}{\mathrm{tr}(P^\top f(X)L_b f^\top(X)P)}$ obtained with P in previous trace-ratio problem is the optimized score for feature selection criterion.

C. Optimization

We consider solving the proposed objective function in Eq.(8) by alternating optimization approach, due to the complex non-linearity of the encoder and decoder, to iteratively update the auto-encoder parameters W_1, W_2, b_1, b_2 and feature selection variable P as well as γ. The first sub-problem is feature selection score learning and then the learned feature selection matrix and score are used in the regularized auto-encoder optimization.

1) Feature Selection Score Learning: In the first sub-problem, assume that the other parameters from auto-encoder are constant, we follow the traditional trace-ratio strategy to optimize the feature selection matrix P and the score γ. Specifically, the trace-difference equation has the following form:

$$P=\arg\min_{P} \mathrm{tr}(P^\top f(X)(L_w-\gamma L_b)f^\top(X)P), \tag{9}$$

Suppose P_t is the optimal result in t-th optimization iteration, thus γ_t is calculated by

$$\gamma_t=\frac{\mathrm{tr}(P_t^\top f(X)L_w f^\top(X)P_t)}{\mathrm{tr}(P_t^\top f(X)L_b f^\top(X)P_t)}, \tag{10}$$

Therefore, we can obtain $r(\gamma_t)$ as

$$r(\gamma_t)=\mathrm{tr}(P_{t+1}^\top f(X)(L_w-\gamma_t L_b)f^\top(X)P_{t+1}), \tag{11}$$

where P_{t+1} can be efficiently calculated according to the rank of scores for each single feature. Note that $r(\gamma)$ is piecewise linear since P is not fixed w.r.t γ, and the slope of $r(\gamma)$ at point γ_t is,

$$r'(\gamma_t)=-\mathrm{tr}(P_{t+1}^\top f(X)L_b f^\top(X)P_{t+1}), \tag{12}$$

We use a linear function $q(\gamma)$ to approximate the piecewise linear function $r(\gamma)$ at point γ_t such that,

$$\begin{aligned}q(\gamma)&=r'(\gamma_t)(\gamma-\gamma_t)+g(\gamma_t)\\ &=\mathrm{tr}(P_{t+1}^\top f(X)(L_w-\gamma L_b)f^\top(X)P_{t+1}).\end{aligned} \tag{13}$$

Let $q(\gamma_{t+1})=0$, we have

$$\gamma_{t+1}=\frac{\mathrm{tr}(P_{t+1}^\top f(X)L_w f^\top(X)P_{t+1})}{\mathrm{tr}(P_{t+1}^\top f(X)L_b f^\top(X)P_{t+1})}. \tag{14}$$

Since $q(\gamma)$ approximates $r(\gamma)$, γ_{t+1} in Eq.(14) is an approximation to the root of equation $r(\gamma)=0$. Updating γ_t by γ_{t+1}, we can obtain an iterative procedure to find the root of equation $r(\gamma)=0$ and thus the learned feature selection matrix P in Eq.(5). One thing to be noted that γ is a fixed parameter passed to the following auto-encoder optimization as a learned global optimal score for the feature selection criterion. **Algorithm 1** summarizes the learning procedure. More details on globally optimal solution and its proof for above trace-ratio problem could be referred to [18].

Algorithm 1 Feature Selection Score Optimization

Input: Learned hidden layer feature $f(X)$, selected feature number m, matrices L_w and L_b
Initialize: $P=I$, $I\in\mathbb{R}^{z\times m}$ is the identity matrix, $\gamma=\frac{\mathrm{tr}(P^\top f(X)L_w f^\top(X)P)}{\mathrm{tr}(P^\top f(X)L_b f^\top(X)P)}$, $\epsilon=10^{-9}$, $iter=0$, $maxiter=20$
while *not converge and iter* $\leq maxiter$ **do**
  Calculate the score of each j-th feature with Eq.(10) by setting $P=[\underbrace{0,...,0}_{j-1},1,\underbrace{0,...,0}_{z-j}]^\top$.
  Rank the features according to the scores in ascending order
  Select the leading m features to update $P\in\mathbb{R}^{z\times m}$
  Calculate $\gamma=\frac{\mathrm{tr}(P^\top f(X)L_w f^\top(X)P)}{\mathrm{tr}(P^\top f(X)L_b f^\top(X)P)}$ with Eq.(10)
  Check the convergence conditions: $\|\gamma_{old}-\gamma\|<\epsilon$
end
Output: feature selection matrix P, global optimal trace ratio score γ

2) Regularized Auto-Encoder Optimization: as the feature selection matrix P and score γ are learned from above, the stochastic sub-gradient descent method can be employed to obtain the parameters W_1, b_1, W_2 and b_2. The gradients of $\mathcal{L}$ in Eq.(8) with respect to the decoding parameters are derived as follows:

$$\frac{\partial\mathcal{L}}{\partial W_2}=(X-g(f(X)))\odot\frac{\partial g(f(X))}{\partial W_2}f^\top(X), \tag{15}$$

$$\frac{\partial\mathcal{L}}{\partial B_2}=(X-g(f(X)))\odot\frac{\partial g(f(X))}{\partial W_2}=\mathcal{L}_2, \tag{16}$$

$$\frac{\partial\mathcal{L}}{\partial W_1}=(W_2^\top\mathcal{L}_2+\lambda P P^\top f(X)(L_w-\gamma L_b))\odot\frac{\partial f(X)}{\partial W_2}X^\top, \tag{17}$$

$$\frac{\partial\mathcal{L}}{\partial B_1}=(W_2^\top\mathcal{L}_2+\lambda P P^\top f(X)(L_w-\gamma L_b))\odot\frac{\partial f(X)}{\partial W_2}. \tag{18}$$

Algorithm 2 Alternating Solve Eq.(8)

Input: Training data X, Parameters λ, layersize, select feature number $m < z$
Initialize: W_1, W_2, b_1 and b_2 are initialized with original auto-encoder, maxiter = 20, iter=0, $\epsilon = 10^{-7}$
while *not converged and* iter $\leq$ maxiter **do**
  Fix others and update P and γ using Eq. (9);
  Fix P and update W_1, W_2, b_1 and b_2 with Eq. (19);
  Check the convergence conditions: $\|\mathcal{L}_{new} - \mathcal{L}_{old}\|_\infty < \epsilon$
end
Output: $W_1, W_2,\ b_1,\ b_2,\ P$ ($Y = P^\top \sigma(W_1 X + B_1)$ could be used as input of next DFSE, to form stack architecture)
Testing: new feature represented with: $Y_{test} = P^\top \sigma(W_1 X_{test} + B_1)$

Then, W_1, W_2 and b_1, b_2 can be updated by using the gradient descent algorithm as follows:

$$W_1 = W_1 - \eta \frac{\partial \mathcal{L}}{\partial W_1}, \quad b_1 = b_1 - \eta \frac{\partial \mathcal{L}}{\partial b_1},$$
$$W_2 = W_2 - \eta \frac{\partial \mathcal{L}}{\partial W_2}, \quad b_2 = b_2 - \eta \frac{\partial \mathcal{L}}{\partial b_2}, \tag{19}$$

where η is the learning rate. $\frac{\partial \mathcal{L}}{\partial b_1}$ and $\frac{\partial \mathcal{L}}{\partial b_2}$ are the column mean of $\frac{\partial \mathcal{L}}{\partial B_1}$ and $\frac{\partial \mathcal{L}}{\partial B_2}$, accordingly. **Algorithm 2** summarizes the details of the iteratively updating of the above two sub-problems.

D. Examples of Supervised and Unsupervised Scenarios

Previous model is a more general one, which can fit to different scenarios by adapting L_w and L_b according to the real situation. In this paper, we adopt Fisher score [14], [18] as the supervised feature selection criterion for classification, while Laplacian score [16] as the unsupervised feature selection method for clustering task. Note that there is no restriction about which method could be integrated in, either Fisher score [14] or other criterions are all suitable. We choose Fisher and Laplacian score here due to their good reported performance and wide usage.

In general, Fisher and Laplacian score could be intergraded into graph embedding framework. For Fisher score, we construct two weighted undirected graphs G_w and G_b on given data (the original input data X is used in this paper to preserve the geometric structure during selecting features), which respectively reflect the within- and between-class affinity relationship [24]. Correspondingly, two weighted matrices S_w and S_b are produced to characterize two graphs respectively. Therefore, the Laplacian matrix defined as $L_w = D_w - S_w$ is obtained, where D_w is the diagonal matrix of S_w, similar for L_b and S_b.

Laplacian Score is essentially based on Laplacian Eigenmaps [25]. For each feature, the Laplacian score is computed to reflect its locality preserving ability. We construct a nearest neighbor graph G in order to model the local geometric structure. The weighted matrix S is constructed by putting $S_{ij} = e^{-\|x_i - x_j\|^2/\tau}$ between k nearest neighbors notes pair x_i and x_j, while setting as 0 otherwise, where τ is a suitable constant. Then, the graph Laplacian matrix L is calculated as $L = D - S$, where D is the diagonal matrix of S. Similar as Fisher score, the task now is to seek the feature selection matrix P with the minimum Laplacian score. Therefore, we can easily set $L_w = L$ and $L_b = D$ to build an unsupervised feature selection guided auto-encoder framework for clustering.

E. Relations to Existing Methods

The proposed algorithm aims to jointly select most important features through the non-linear dimensionality reduction. Here we elaborate a few connections with existing methods, and therefore highlight the advantage of our model.

1) Sparse AE: Sparse Auto-Encoders (SAE) aims to select a small set of hidden units to reconstruct the input by enforcing the activation of the majority hidden units to be close to 0 [26]. In a different manner, the proposed DSFE framework desires to seek a small set of hidden units which preserve most discerning information for later classification or clustering tasks, while other hidden units would be still used for reconstruction. In other words, the important discerning hidden units and task-irrelevant units are treated separately. The key idea in our methods is that, we need to endow the specific hidden units with more discriminative ability while not all of the units, and left the others only focus on reconstruction.

2) Embedding With AE Regularization: Yu *et al.* [10] proposed an embedding with auto-encoder regularization (EAER) framework, whose idea is to utilize graph embedding structure to guide the auto-encoder reconstruction. Specifically, they want to use graph regularization to preserve more neighborhood relationships between the data in the input space during the non-linear dimensionality reduction. Although our proposed DSFE framework also adopts graph regularizer, the idea of ours is to preserve the geometric structure of the data during the feature selection on hidden units, which is jointly optimized with the non-linear dimensionality reduction.

IV. Experiments

In this section, we evaluate the proposed DSFE method through both classification and clustering schemes. For classification problem, we adopt two benchmark datasets, including one object dataset (COIL100 [38]) and one face dataset (CMU-PIE [39]). The clustering experiments are performed on 9 real-world image datasets (Table.III summarizes some important characteristics). We first introduce the compared algorithms, experimental setting, then showcase the effectiveness of our DSFE on object classification, face recognition and image clustering. Note that our DSFE model with Fisher score regularizer (DSFE$_\mathcal{F}$) is evaluated in supervised classification, while the one with Laplacian score regularizer (DSFE$_\mathcal{L}$) is tested in unsupervised clustering. Experimental results will be presented with analysis in this section. What's more, visualization tasks are also deployed to intuitively demonstrate our method's ability to extract discerning units.

3) Comparative Algorithms for Classification: Several manifold learning methods: LDA [4], LRC [27], SRRS [28], LCLRD [29]; and four auto-encoder based methods: sparse

TABLE I

AVERAGE RECOGNITION RATE(%) WITH STANDARD DEVIATIONS OF DIFFERENT METHODS ON COIL-100 DATABASE WITH DIFFERENT NUMBER OF CLASSES. BOLD DENOTES THE BEST RESULTS

Methods	20 objects	40 objects	60 objects	80 objects	100 objects	Average
LDA [4]	81.94 ± 1.21	76.73 ± 0.30	66.16 ± 0.97	59.19 ± 0.73	52.48 ± 0.53	67.30
LRC [27]	90.74 ± 0.71	89.00 ± 0.46	86.57 ± 0.37	85.09 ± 0.34	83.16 ± 0.64	86.91
SRRS [28]	92.03 ± 1.21	92.51 ± 0.65	90.82 ± 0.43	88.75 ± 0.71	85.12 ± 0.33	89.85
LCLRD [29]	92.15 ± 0.34	89.86 ± 0.49	87.23 ± 0.29	85.40 ± 0.61	84.15 ± 0.39	87.75
NPE [30]	82.24 ± 2.25	76.01 ± 1.04	63.22 ± 1.36	52.18 ± 1.44	30.73 ± 1.31	60.88
LSDA [31]	82.79 ± 1.70	75.01 ± 1.14	62.85 ± 1.41	51.69 ± 2.05	26.77 ± 1.05	59.82
FDDL [32]	85.74 ± 0.77	82.05 ± 0.40	77.22 ± 0.74	74.81 ± 0.55	73.55 ± 0.63	78.67
DLRD [33]	88.61 ± 0.95	86.39 ± 0.54	83.46 ± 0.15	81.50 ± 0.47	79.91 ± 0.59	83.97
$D^2L^2R^2$[34]	90.98 ± 0.38	88.27 ± 0.38	86.36 ± 0.53	84.69 ± 0.45	83.06 ± 0.37	86.67
DPL [35]	87.55 ± 1.32	85.05 ± 0.21	81.22 ± 0.21	78.78 ± 0.85	76.28 ± 0.94	81.77
SAE [26]	91.28 ± 0.68	89.65 ± 0.77	87.26 ± 0.85	85.90 ± 0.61	84.46 ± 0.61	87.71
mDAE [36]	91.10 ± 0.72	88.82 ± 0.68	86.86 ± 0.83	85.52 ± 0.62	83.66 ± 0.56	87.19
GAE [10]	93.37 ± 0.80	91.33 ± 0.58	89.11 ± 0.46	86.67 ± 0.57	85.96 ± 0.43	89.28
mGAE [37]	93.06 ± 1.12	90.81 ± 0.87	88.06 ± 0.97	85.88 ± 0.66	84.77 ± 0.78	88.52
AE+FS [ours]	91.73 ± 0.75	90.24 ± 0.89	87.98 ± 0.81	86.42 ± 0.57	85.05 ± 0.60	88.28
$DSFE_{\mathcal{F}}$ [ours]	**94.12 ± 0.45**	**93.82 ± 0.60**	**91.97 ± 0.94**	**89.68 ± 0.81**	**86.75 ± 0.79**	**91.27**

auto-encoder (SAE) [26], marginalized denoising auto-encoders (mDAE) [36], graph regularized auto-encoder (GAE) [10] and marginalized graph auto-encoder (mGAE) [37] are used as baseline algorithms. What's more, for verifying the advantage of joint learning, we propose a simple combination framework as comparison, named as AE+FS, which first uses a traditional auto-encoder to learn a new representation, then the same trace-ratio feature selection procedure is applied on the obtained hidden layer to produce a subset of features. In other words, it is a separate learning version of our algorithm. Besides above baseline methods, several different state-of-the-art algorithms are compared in each dataset to show our advantage. Note that the layer-size setting for those AE based methods are all the same and they only differ in the regularizer.

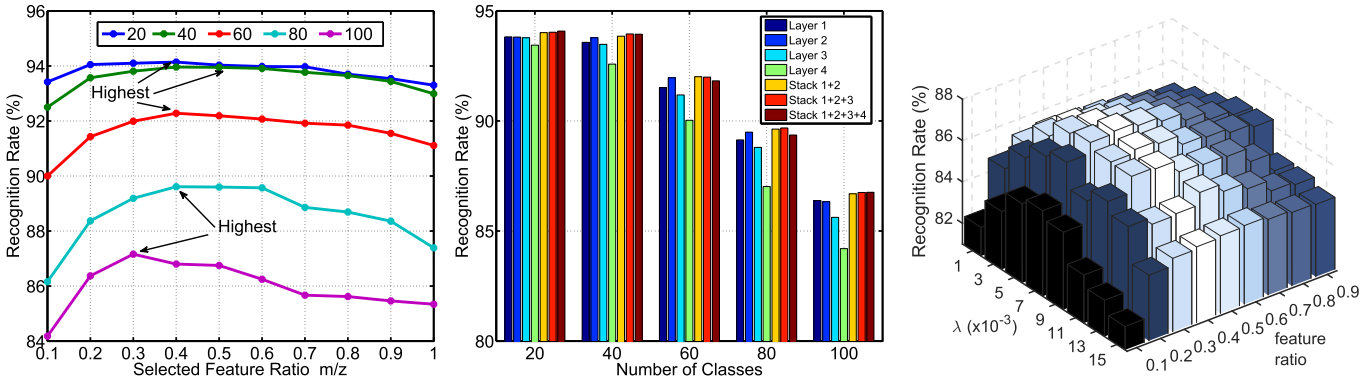
4) Comparative Algorithms for Clustering: Except K-means and sparse auto-encoder (SAE) [26] are adopted as baseline methods, the simple combination framework AE+FS is also used as comparison. What's more, to validate the effectiveness of our model, we compare with several state-of-the-art methods in terms of ensemble clustering methods (e.g., EAC [40], SEC [41]), and deep clustering (e.g., MAEC [36], GEncoder [42], and DLC [43]). Note that the deep clustering methods are set with 5 layers.

5) Parameter Selection: In addition to the parameter of auto-encoder (layer-size setting), there are two more parameters in the objective function (Eq.(8)), which are the feature selection ratio (m/z) and the balance parameter λ. One thing to be noted that γ is automatically learned as global optimal score in feature selection step. The selected feature size is set to 50% of the original hidden layer size for all the experiments, and we will analyze the impact of selection ratio. λ balances the feature selection regularization and the loss function of AE, we empirically set it in our experiments and will give analysis in following sections. Specifically, λ is set as 2×10^{-3} for COIL100, 3×10^{-3} for CMU-PIE and 2×10^{-3} for all datasets in clustering experiments. For construction of the nearest neighbor graph in $DSFE_{\mathcal{L}}$, the number of nearest neighbor nodes k is set to 5 for all datasets.

A. Classification

COIL100 consists 100 objects each has 72 color images (totally 7,200 images) with different lighting conditions. Each object was collected in equally spaced views at every 5 degrees of rotation. The images are converted to gray scale, and normalized to 32×32. Each object is randomly selected 10 images to form the training set while the rest consist the testing. The random training and test split is repeated 20 times. We separately deploy experiments on subset of 20, 40, 60, 80 and 100 objects to evaluate the scalability of the testing methods. Several subspace methods and latest DL based classification methods i.e. NPE [30], LSDA [31], FDDL [32], DLRD [33], $D^2L^2R^2$ [34], and DPL [35] are compared in this dataset. Each compared method is either tuned to achieve their best performance or directly copied from the original papers under same experimental setting. We set the layer as [300, 200, 100] for all all AE based methods, and report the results on the feature with three layer stack together.

Table.I summarizes the average recognition rates of all compared algorithms. The results show a large improvement by our algorithm. Fig.2 shows the analysis of $DSFE_{\mathcal{F}}$ with different values of selected feature ratio and layer-size setting, respectively. It can be seen from Fig.2(a) that the highest recognition rate appears mostly when the selected feature number is 30%∼50% of original feature size. For this reason, the ratio is set to 0.5 in all of our experiments for simplicity. We use different layer of a stacked $DSFE_{\mathcal{F}}$ to test recognition rate in Fig.2(b). Four layers of $DSFE_{\mathcal{F}}$ are set with layer-size [300, 200, 100, 50]. It can be observe from the figure that layer 2 itself or all layers stacked together perform the best. The reason could be higher layer has the input with more discerning units selected, and the encoder could focus more on representing the task-relevant information. The drops at layer 3 and 4 are probably due to the too few units to capture



(a) (b) (c)

Fig. 2. (a) Recognition rates with different subset of COIL-100 in terms of feature selection ratio (m/z). The highest results appear mostly at 0.3∼0.5. (b) Recognition rates with different layer combination used on COIL-100. (c) Impact of feature selection ratio (m/z) and λ on COIL-100.

TABLE II

AVERAGE RECOGNITION RATE(%) WITH STANDARD DEVIATIONS OF DIFFERENT METHODS ON CMU PIE DATABASE WITH DIFFERENT NUMBER OF TRAINING SAMPLES EACH CLASSES. BOLD DENOTES THE BEST RESULTS

Methods	5Train	10Train	20Train	30Train	40Train	50Train	60Train
LDA [4]	57.18 ± 1.28	69.31 ± 0.82	78.51 ± 0.49	89.09 ± 0.31	92.19 ± 0.37	93.72 ± 0.14	94.57 ± 0.13
LRC [27]	40.51 ± 1.04	68.81 ± 0.77	86.62 ± 0.63	91.85 ± 0.48	93.86 ± 0.46	95.09 ± 0.21	95.88 ± 0.18
SRRS [28]	60.02 ± 1.23	70.38 ± 1.31	80.17 ± 0.61	89.24 ± 0.32	92.38 ± 0.43	93.86 ± 0.31	94.93 ± 0.21
LCLRD [29]	67.50 ± 1.07	84.10 ± 0.89	89.91 ± 0.78	92.38 ± 0.47	94.12 ± 0.37	94.19 ± 0.23	95.29 ± 0.18
SAE [26]	69.33 ± 1.50	80.91 ± 1.10	88.99 ± 0.62	91.68 ± 0.36	93.02 ± 0.34	93.73 ± 0.31	94.08 ± 0.24
mDAE [36]	67.04 ± 1.26	80.94 ± 0.83	88.24 ± 0.78	91.58 ± 0.42	92.72 ± 0.31	93.28 ± 0.30	93.58 ± 0.30
GAE [10]	71.65 ± 1.28	84.06 ± 0.85	90.38 ± 0.77	92.32 ± 0.44	93.22 ± 0.30	93.74 ± 0.30	94.04 ± 0.24
mGAE [37]	71.47 ± 1.26	83.92 ± 0.75	90.50 ± 0.64	93.16 ± 0.33	93.94 ± 0.23	94.22 ± 0.26	94.35 ± 0.26
AE+FS [ours]	71.47 ± 1.42	83.69 ± 0.87	89.97 ± 0.56	93.19 ± 0.37	94.18 ± 0.29	94.78 ± 0.34	95.02 ± 0.27
DSFE$_{\mathcal{F}}$ [ours]	**72.43 ± 1.32**	**85.26 ± 0.74**	**91.93 ± 0.57**	**94.47 ± 0.30**	**95.73 ± 0.24**	**96.48 ± 0.26**	**96.79 ± 0.18**

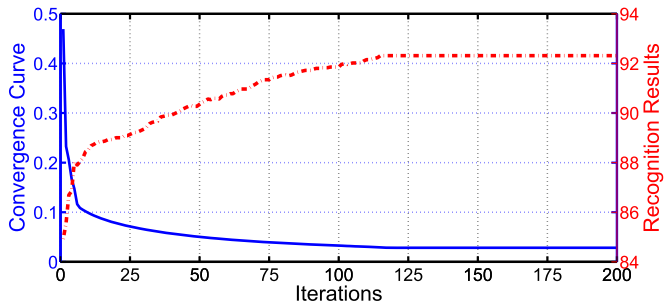


Fig. 3. The optimization process of DSFE$_{\mathcal{F}}$ on the COIL100 dataset with 60 objects.

sufficient information. The performance along with both feature ratio and λ is evaluated in Fig.2(c). The convergence test of our proposed algorithm is also provided in Fig.3.

CMU-PIE is a widely used dataset which contains a total of 41,368 images of 68 subjects, each person has 13 different poses, 43 different illumination conditions, and 4 different expressions. From all the 13 different poses, a subset of five near frontal poses (C05, C07, C09, C27, C29) are selected, with all the faces under different illuminations and expressions. In this way, a subset with 11,554 images is used in our experiment and each person has about 170 images. We repeat each experiment 20 times with random training and test split. Table. II summarizes the recognition rates with different numbers of training samples selected from per person. All images are normalized to 32×32 and the model is single-layer with dimension of 1500. The result indicates our method can not only work on object categorization but also on face recognition. Besides, the impact of feature selection ration and λ are reported in Fig.5. The highest result is given when selection ratio $= 0.55$ and $\lambda = 3 \times 10^{-3}$.

B. Clustering

In order to demonstrate the effectiveness of our DSFE$_{\mathcal{L}}$ model, the evaluated datasets are selected with different levels of features, types of data, numbers of feature and instances. As shown in Table III, the first dataset is low-level feature for character[1], the middle ones are the object[2] and digit datasets[3] with Surf and raw pixel features. The last two datasets are objects with deep learning features[4,5]. Fig. 4 shows some sample images from these datasets. The compared ensemble clustering methods and deep clustering methods are summarized as follow:

- EAC [40] is an evidence accumulation-based clustering algorithm which applies hierarchical agglomerative clustering on the co-association matrix.

[1]http://archive.ics.uci.edu/ml/
[2]https://people.eecs.berkeley.edu/~jhoffman/domainadapt/
[3]http://www.cad.zju.edu.cn/home/dengcai/
[4]http://www.cs.dartmouth.edu/~chenfang/
[5]http://groups.csail.mit.edu/vision/SUN/

Fig. 4. Sample Images. (a) The USPS is a 0-9 handwritten digits database in grey level, (b) DSLR contains office environment images taken with varying lighting and pose changes using a dslr camera, and (c) SUN09 is a five categories subset from SUN dataset which covering a large variety of scenes, places and the objects within. Here we show some samples in category 'cars'.

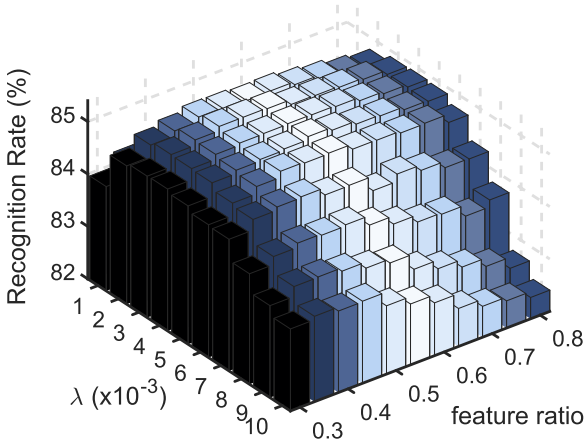


Fig. 5. Impact of feature selection ratio (m/z) and λ on the 10 train CMU PIE dataset in terms of classification accuracy.

TABLE III
EXPERIMENTAL DATA SETS FOR CLUSTERING

Dataset	Type	Feature	#Instance	#Feature	#Class
letter	character	low-level	20000	16	26
COIL100	object	middle-level	7200	1024	100
Amazon	object	middle-level	958	800	10
Caltech	object	middle-level	1123	800	10
Dslr	object	middle-level	157	800	10
Webcam	object	middle-level	295	800	10
USPS	digit	middle-level	9298	256	10
SUN09	object	high-level	3238	4096	5
VOC2007	object	high-level	3376	4096	5

- SEC [41] finds the consensus partition by employing spectral clustering on co-association matrix. It equivalently results in weighted K-means clustering.
- MAEC [36] applies marginalized denoising auto-encoders (mDAE) on the Laplacian graph as the new input representations and gets the partition with K-means.
- GEncoder [42] feeds the graph Laplacian matrix in the sparse auto-encoder to get the graph representations.
- DLC [43] is short for deep linear coding, which jointly learns a linear transform and discriminative codings at the same time in a deep structure.

Validation Metric: We introduce five external metrics to measure the performance, which are accuracy, the normalized Mutual Information (MI_n), the normalized Variation of Information (VI_n), the normalized van Dongen criterion (VD_n) and the normalized Rand statistic (R_n). Note that accuracy, MI_n and R_n are positive measurements (the larger, the better), while VI_n and VD_n are negative measurements (the smaller, the better). The computation details for these metrics can be found in [44]. The comparative results are reported in accuracy and MI_n, while the R_n, VI_n and VD_n are further used in parameter analysis.

Tables. IV and V show the performance of different clustering algorithms measured by accuracy and MI_n respectively. Note that each method is tested with 20 times random K-means initialization and the average performance is reported. 'N/A' means MAEC and GEncoder can't handle large-scale datasets due to the high computational cost. Our method gets the best results on most of 9 datasets, even outperforms the stat-of-the-art ensemble clustering methods. It is worth to mention that the improvements are nearly 15% on Amazon, Dslr, and Webcam datasets. Fig.6 shows the analysis of one layer $\mathrm{DSFE}_{\mathcal{L}}$ with different values of selected feature ratio on Amazon, Caltech, Dslr and Webcam datasets in terms of five metrics.

C. Visualization

In addition, we provide feature embedding visualization in a two-dimensional space by applying the t-SNE algorithm [45]. Two widely-used datasets, i.e., VOC2007 and ImageNet, are used for visualization. We apply the same VOC2007 dataset as in clustering experiment, and the subset of ImageNet we use here contains 7,341 samples in 5 categories with 4096-dim features. Half of the dataset is used to train our $\mathrm{DSFE}_{\mathcal{F}}$ model and applied on another half test data. The learned features of test data is then mapped to a two-dimensional space with t-sne. The results in Fig. 7 validate that comparing to the mapping of original data, by applying our model we can obtain more discerning features. Another visualization result could be found in Fig. 1. The left column images are the input sample from YaleB and ARface datasets, while the right column images indicate the region where its reconstruction involves more of our learned discerning feature. In other words, those highlighted regions are concerned as more useful on future classification tasks, as they are coded more on the selected task-relevant features. As we can observe, the eyes, nose, mouth, face contour, beard are the most focused regions on the first two rows images. For the lower two rows, the mouth region is been neglected since the mouth movement in the

TABLE IV

CLUSTERING PERFORMANCE OF DIFFERENT ALGORITHMS MEASURED BY ACCURACY. BOLD DENOTES THE BEST RESULTS

Dataset	Baseline	Ensemble Clustering Method		Deep Clustering Method					
	K-means	HCC [40]	SEC [41]	SAE [26]	MAEC [36]	GEncoder [42]	DLC [43]	AE+FS[ours]	DSFE$_{\mathcal{L}}$[ours]
letter	0.2485	0.2447	0.2137	0.3070	N/A	N/A	0.3087	0.2492	**0.3193**
COIL100	0.5056	0.5332	0.5210	0.5062	0.5206	0.0103	**0.5348**	0.5261	0.5271
Amazon	0.3309	0.3069	0.3424	0.3761	0.2443	0.2004	0.3653	0.3584	**0.4134**
Caltech	0.2457	0.2386	0.2680	0.2683	0.2102	0.2333	0.2840	0.2900	**0.2967**
Dslr	0.3631	0.3949	0.4395	0.4981	0.3185	0.2485	0.4267	0.4764	**0.5159**
Webcam	0.3932	0.3932	0.4169	0.4384	0.3220	0.3430	0.5119	0.5144	**0.5191**
USPS	0.6222	0.6137	0.6157	0.6563	0.4066	0.1676	0.6457	0.6685	**0.6920**
SUN09	0.4360	0.4235	0.4732	0.4766	0.3696	0.3854	0.4829	0.4831	**0.4859**
VOC2007	0.4565	0.5044	0.5124	0.4914	0.3874	0.4443	**0.5130**	0.4959	0.5100

TABLE V

CLUSTERING PERFORMANCE OF DIFFERENT ALGORITHMS MEASURED BY MI$_n$. BOLD DENOTES THE BEST RESULTS

Dataset	Baseline	Ensemble Clustering Method		Deep Clustering Method					
	K-means	HCC [40]	SEC [41]	SAE [26]	MAEC [36]	GEncoder [42]	DLC [43]	AE+FS[ours]	DSFE$_{\mathcal{L}}$[ours]
letter	0.3446	0.3435	0.3090	0.4063	N/A	N/A	0.3977	0.3424	**0.4133**
COIL100	0.7719	**0.7815**	0.7786	0.7757	0.7794	0.0924	0.7764	0.7788	0.7781
Amazon	0.3057	0.3062	0.2595	0.3283	0.1982	0.0911	0.3001	0.3239	**0.3618**
Caltech	0.2043	0.2094	0.1979	0.2026	0.1352	0.1132	0.2104	0.2068	**0.2128**
Dslr	0.3766	0.4776	0.4756	0.4554	0.2900	0.1846	0.4614	0.4411	**0.5009**
Webcam	0.4242	0.4565	0.4441	0.4913	0.2316	0.3661	0.5280	0.5176	**0.5381**
USPS	0.6049	0.5187	0.5895	0.6248	0.4408	0.0141	0.5843	0.6217	**0.6270**
SUN09	0.2014	0.2091	0.1927	0.2113	0.0576	0.0481	0.2315	0.2162	**0.2364**
VOC2007	0.2697	0.2564	0.2511	0.2917	0.1118	0.1920	0.2651	0.2949	**0.3092**

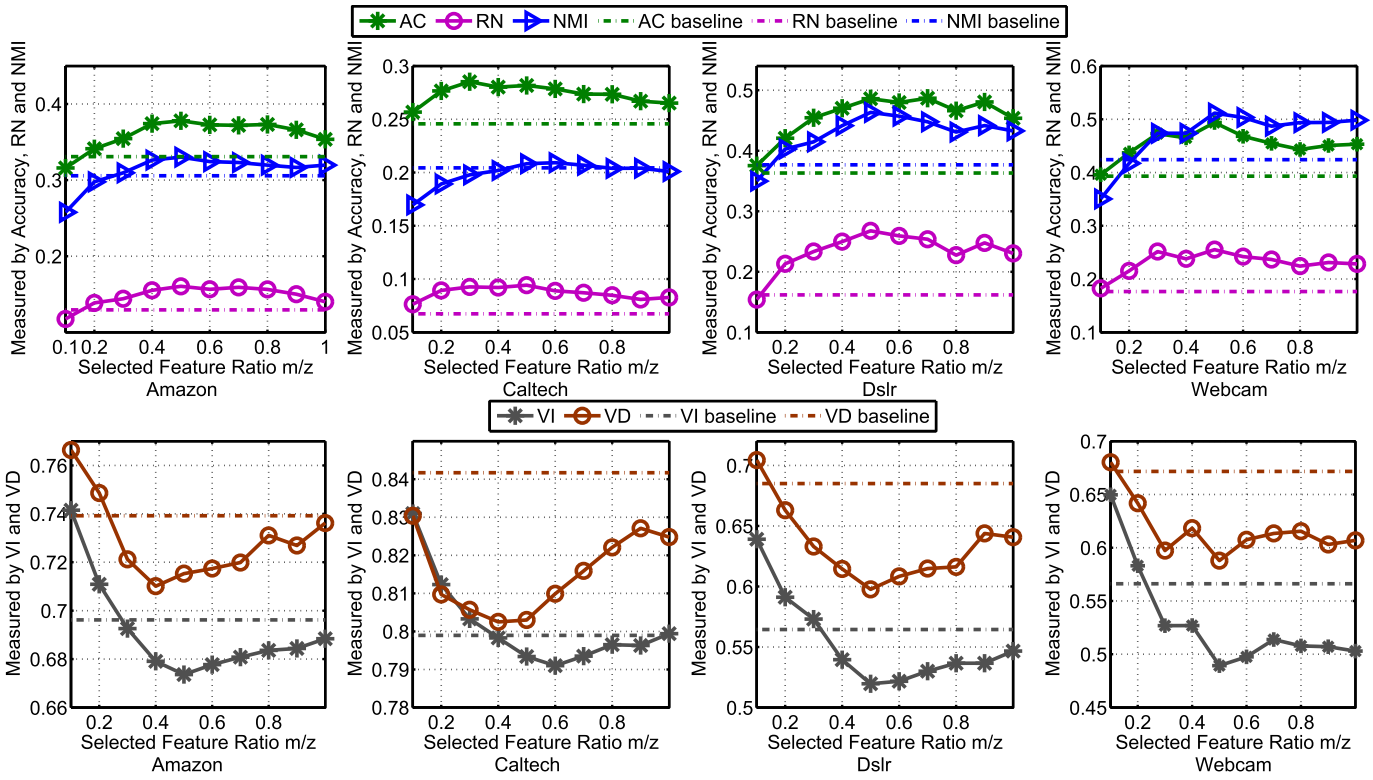
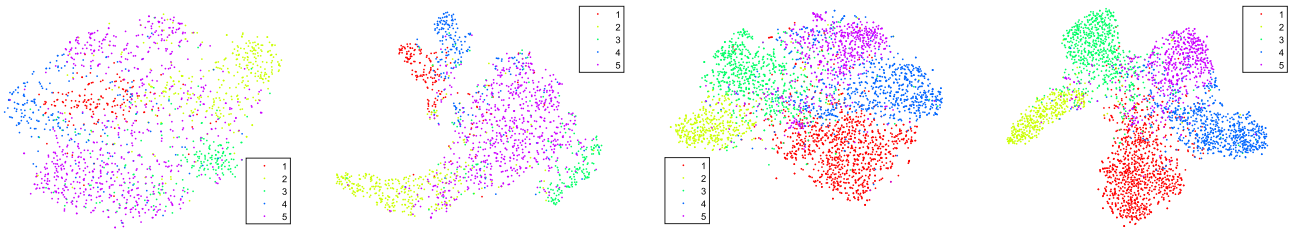


Fig. 6. Impact of feature selection ratio (m/z) on Amazon, Caltech, Dslr and Webcam datasets in terms of five measurements. The first row are measured by accuracy, R$_n$ and MI$_n$ which are positive measurements while the second row are VI$_n$ and VD$_n$, which are negative measurements. Each measurement's baseline is produced with directly K-means.

ARface dataset works as a variable factors which might mislead the classification.

More reconstruction results with selected features are reported in Fig. 8. We show two identities' samples from ARface dataset, each has four input samples list in left column with expressions and different lighting conditions. The middle images are reconstructed from all hidden units which has good reconstruction for each correspondence factors. While the right



(a) (b) (c) (d)

Fig. 7. Visualization of features learned from our model of the VOC2007 and ImageNet datasets by t-SNE. From left to right, the corresponding figures are for (a) original data, (b) learned features from VOC2007 dataset, and (c) original data, (d) learned features from the subset of ImageNet dataset. Different colors represent different categories.

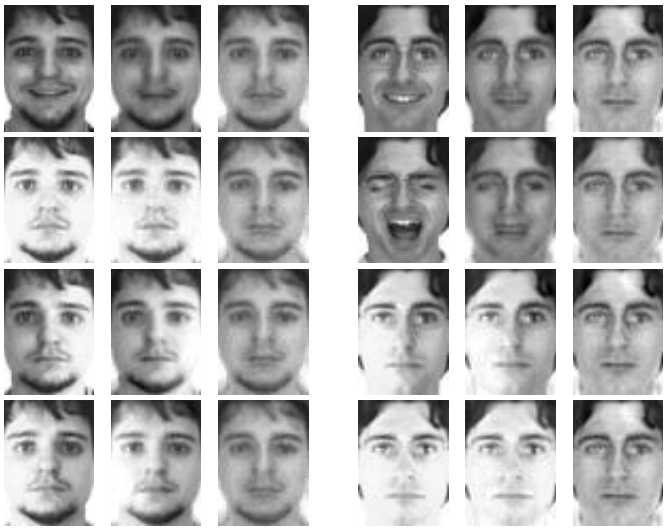


Fig. 8. Visualization of the reconstruct image with different hidden-layer units on ARface dataset. There are samples for two identities, each has four input samples in left column. The middle and right columns are the reconstructed output with whole hidden units and only use selected discerning units respectively.

columns images are reconstructed from only those selected discerning units, which we can observe that the reconstructed images are more consistency and have eliminated the task-irrelevant factors, i.e., expression and lighting conditions.

V. Conclusion

In this work, a unified framework was introduced to jointly train non-linear transformation and select high-level informative features targeting both supervised and unsupervised schemes. Instead of endowing all the hidden units with discriminative ability like other existing graph based auto-encoder, we distinguish the hidden units contain discerning information from those task-irrelevant ones. The regularizer applied on the selected hidden layer features in turn compresses the discriminative information only on the selected task-relevant units. As a general framework, different feature selection criterions are fitted into our DSFE model and evaluated on different tasks. The supervised recognition results on three benchmark datasets and the evaluation on unsupervised clustering tasks both indicated the effectiveness of the proposed DSFE framework.

References

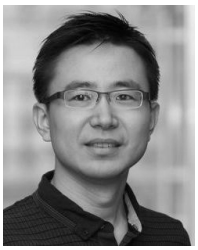
[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Hoboken, NJ, USA: Wiley, 2001.
[2] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.
[3] L. C. Molina, L. Belanche, and A. Nebot, "Feature selection algorithms: A survey and experimental evaluation," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2002, pp. 306–313.
[4] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.
[5] M. Shao, C. Castillo, Z. Gu, and Y. Fu, "Low-rank transfer subspace learning," in *Proc. IEEE 12th Int. Conf. Data Mining*, Dec. 2012, pp. 1104–1109.
[6] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Comput. Graph. Statist.*, vol. 15, no. 2, pp. 265–286, 2006.
[7] Q. Gu, Z. Li, and J. Han, "Joint feature selection and subspace learning," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1294–1299.
[8] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[9] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Mar. 2010.
[10] W. Yu, G. Zeng, P. Luo, F. Zhuang, Q. He, and Z. Shi, "Embedding with autoencoder regularization," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Berlin, Germany: Springer, 2013, pp. 208–223.
[11] L. Wang, T. Liu, G. Wang, K. L. Chan, and Q. Yang, "Video tracking using learned hierarchical features," *IEEE Trans. Image Process.*, vol. 24, no. 4, pp. 1424–1435, Apr. 2015.
[12] C. Hong, J. Yu, J. Wan, D. Tao, and M. Wang, "Multimodal deep autoencoder for human pose recovery," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5659–5670, Dec. 2015.
[13] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2011, pp. 151–161.
[14] Q. Gu, Z. Li, and J. Han. (2012). "Generalized Fisher score for feature selection." [Online]. Available: https://arxiv.org/abs/1202.3725
[15] H. Liu and H. Motoda, *Computational Methods of Feature Selection*. Boca Raton, FL, USA: CRC Press, 2007.
[16] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. 18th Int. Conf. Neural Inf. Process. Syst.*, 2005, pp. 507–514.
[17] L. Song, A. Smola, A. Gretton, K. M. Borgwardt, and J. Bedo, "Supervised feature selection via dependence estimation," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 823–830.
[18] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan, "Trace ratio criterion for feature selection," in *Proc. 23rd Nat. Conf. Artif. Intell.*, vol. 2, 2008, pp. 671–676.
[19] M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi, "Domain generalization for object recognition with multi-task autoencoders," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2551–2559.
[20] Z. Ding and Y. Fu, "Robust transfer metric learning for image classification," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 660–670, Feb. 2017.

[21] S. Wang, Z. Ding, and Y. Fu, "Coupled marginalized auto-encoders for cross-domain multi-view learning," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2125–2131.
[22] S. Wang, Z. Ding, and Y. Fu, "Feature selection guided auto-encoder," in *Proc. Assoc. Adv. Artif. Intell. Conf. Artif. Intell.*, 2017, pp. 2725–2731.
[23] S. Yan and X. Tang, "Trace quotient problems revisited," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2006, pp. 232–244.
[24] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.
[25] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. 14th Int. Conf. Neural Inf. Process. Syst.*, 2001, pp. 585–591.
[26] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 215–223.
[27] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 11, pp. 2106–2112, Nov. 2010.
[28] S. Li and Y. Fu, "Learning robust and discriminative subspace with low-rank constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2160–2173, Nov. 2016.
[29] S. Wang and Y. Fu, "Locality-constrained discriminative learning and coding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 17–24.
[30] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Proc. Tenth IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 1, Oct. 2005, pp. 1208–1213.
[31] D. Cai, X. He, K. Zhou, J. Han, and H. Bao, "Locality sensitive discriminant analysis," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, 2007, pp. 708–713.
[32] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 543–550.
[33] L. Ma, C. Wang, B. Xiao, and W. Zhou, "Sparse representation for face recognition based on discriminative low-rank dictionary learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2586–2593.
[34] L. Li, S. Li, and Y. Fu, "Learning low-rank and discriminative dictionary for image classification," *Image Vis. Comput.*, vol. 32, no. 10, pp. 814–823, 2014.
[35] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Projective dictionary pair learning for pattern classification," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 793–801.
[36] M. Chen, Z. Xu, K. Weinberger, and F. Sha. (2012). "Marginalized denoising autoencoders for domain adaptation." [Online]. Available: https://arxiv.org/abs/1206.4683
[37] C. Wang, S. Pan, G. Long, X. Zhu, and J. Jiang, "MGAE: Marginalized graph autoencoder for graph clustering," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 889–898.
[38] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (coil 100)," Dept. Comput. Sci., Columbia Univ., New York, NY, USA, Tech. Rep. CUCS-006-96, 1996.
[39] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression (PIE) database," in *Proc. 5th IEEE Int. Conf. Automat. Face Gesture Recog.*, May 2002, pp. 53–58.
[40] A. L. N. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 835–850, Jun. 2005.
[41] H. Liu, T. Liu, J. Wu, D. Tao, and Y. Fu, "Spectral ensemble clustering," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 715–724.
[42] F. Tian, B. Gao, Q. Cui, E. Chen, and T.-Y. Liu, "Learning deep representations for graph clustering," in *Proc. Assoc. Adv. Artif. Intell. Conf. Artif. Intell.*, 2014, pp. 1293–1299.
[43] M. Shao, S. Li, Z. Ding, and Y. Fu, "Deep linear coding for fast graph clustering," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 3798–3804.
[44] J. Wu, H. Xiong, and J. Chen, "Adapting the right measures for K-means clustering," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 877–886.
[45] L. Van Der Maaten, "Accelerating t-SNE using tree-based algorithms," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3221–3245, 2014.

Shuyang Wang received the B.Eng. degree in technology and apparatus of measuring and control from Beihang University, Beijing, China, in 2013, and the Ph.D. degree from the Department of Electrical and Computer Engineering, Northeastern University, USA, in 2018.

His current research interests include low-rank matrix recovery, machine learning, and computer vision. He has served as a Reviewer for IEEE journals, such as IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING.

Zhengming Ding (S'14–M'18) received the B.Eng. degree in information security and the M.Eng. degree in computer software and theory from the University of Electronic Science and Technology of China, China, in 2010 and 2013, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, Northeastern University, USA, in 2018. He has been a Faculty Member affiliated with the Department of Computer, Information and Technology, Indiana University–Purdue University Indianapolis, since 2018. His research interests include transfer learning, multi-view learning, and deep learning. He received the National Institute of Justice Fellowship from 2016 to 2018. He was a recipient of the Best Paper Award (SPIE 2016) and the Best Paper Candidate (ACM MM 2017). He is currently an Associate Editor of the *Journal of Electronic Imaging*.



Yun Fu (S'07–M'08–SM'11–F'19) received the B.Eng. degree in information engineering and the M.Eng. degree in pattern recognition and intelligence systems from Xi'an Jiaotong University, China, and the M.S. degree in statistics and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign. He has been an Interdisciplinary Faculty Member affiliated with the College of Engineering and the Khoury College of Computer and Information Sciences, Northeastern University, since 2012. He has published extensively in leading journals, books/book chapters, and international conferences/workshops. His research interests include machine learning, computational intelligence, big data mining, computer vision, pattern recognition, and cyber-physical systems. He is a fellow of IAPR, OSA, and SPIE, a Lifetime Distinguished Member of ACM, a Lifetime Member of AAAI and the Institute of Mathematical Statistics, a member of the ACM Future of Computing Academy, the Global Young Academy, AAAS, and INNS, and a Beckman Graduate Fellow from 2007 to 2008. He received seven Prestigious Young Investigator Awards from NAE, ONR, ARO, IEEE, INNS, UIUC, and Grainger Foundation; nine Best Paper Awards from the IEEE, IAPR, SPIE, and SIAM; and many major Industrial Research Awards from Google, Samsung, and Adobe. He serves as an Associate Editor, Chair, PC Member, and a Reviewer for many top journals and international conferences/workshops. He is currently an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEANING SYSTEMS.