UNSUPERVISED CLUSTERING FOR DISTORTED IMAGE WITH DENOISING FEATURE LEARNING

Qihao Lin, Jinyu Cai and Genggeng Liu

College of Mathematics and Computer Science, Fuzhou University,
Fuzhou, 350116, China

ABSTRACT

High-dimensional of image data is an obstacle for clustering. One of methods to solve it is feature representation learning. However, if the image is distorted or suffers from the influence of noise, the extraction of effective features may be difficult. In this paper, an end-to-end feature learning model is proposed to extract denoising low-dimensional representations from distorted images, and these denoising features are evaluated by comparing with several feature representation methods in clustering task. First, some related works about classical feature learning are introduced. Then the architecture and working mechanism of denoising feature learning model are presented. As the structural characteristics of this model, it can obtain essential information from image to decrease reconstruction error. When facing with corrupted data, it also runs a robust clustering result. Finally, compared to other unsupervised feature learning methods, extensive experiments demonstrate that the obtained feature representations by proposed model run a competitive clustering performance. The low-dimensional representations can replace the original datasets primely.

KEYWORDS

Unsupervised Learning, Feature Representation, Auto-encoder, Clustering

1. INTRODUCTION

In machine learning and data analysis, difficulties in information processing are caused by large dimensions. Meanwhile, if image datasets are distorted or suffer from noise, the extraction of effective features becomes more difficult. Consequently, learning reusable feature representations from a large number of unlabelled datasets has become a research hotspot. High-dimensional images always need a pre-processing such as dimensionality reduction [1]. Feature representation learning is an effective method [2]. Feature learning can be categorized into supervised based and unsupervised based. Supervised based methods have reached remarkable performance. Linear discriminant analysis (LDA) [3] makes the distance between different types of samples larger, and the distance between similar samples smaller. Locality sensitive discriminant analysis (LSDA) [4] belongs to manifold learning algorithm. Its main idea is to maximize the edge of different classes in each local region. However, as the increasing of data and unmarked label, supervised based methods may have an impact on its accuracy.

The emergence of unsupervised feature learning is better solved 'curse of dimensionality' as well as unmarked labels. Unsupervised feature learning is classified into two parts: linear based and non-linear based. Principal component analysis (PCA) is a statistical method [5]. It uses orthogonal transformation to convert a set of variables that may be related into a set of linearly

David C. Wyld et al. (Eds): MLNLP, BDIoT, ITCCMA, CSITY, DTMN, AIFZ, SIGPRO - 2020
pp. 15-27, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101202

uncorrelated variables. The converted set of variables is called the principal component. Locality preserving projections (LPP) builds a graph on the data set. This graph contains the information of the node's neighbours [6]. The algorithm mainly finds an optimal linear approximation when the high-dimensional data depends on the embedding of the low-dimensional manifold in the surrounding space. PCA and LPP are two linear feature learning algorithms. Neighbourhood preserving embedding selects neighbours to reconstruct linear weights for each point. The core of isometric feature mapping (Isomap) is to find and utilize the characteristics of manifold space, introduce geodesic distance and propose corresponding distance calculation [7]. Locally linear embedding (LLE) is a new feature learning algorithm for non-linear data [8]. It can keep the original manifold structure after dimensionality reduction as far as possible. Isometric projection (IsoP) discovers the in-trinsic geometrical structure of data set [9].

In recent years, auto-encoder (AE) and its family are proposed to realize dimensionality reduction and feature learning. Auto-encoder is an unsupervised learning algorithm (the training example is not marked), which uses back propagation algorithm and makes the target value equal to the input value [10]. It is a neural network which contains three layers. The dimension of hidden layer is much smaller than input layer. Sparse auto-encoder (SAE) limits the number of hidden units to learn more useful features [11]. A neuron is active if its output value is close to 1, otherwise it is not active if its output value is close to 0. Variational auto-encoder (VAE) is an important generation model. It proposes a gradient estimation called stochastic gradient variable bayes [12]. The core of adversarial auto-encoder (AAE) is to use a generator and a discriminator for adversary learning [13]. It's a combination of VAE and adversarial network.

The above methods run a great performance on feature extraction. However, when facing with distorted images, existing unsupervised feature learning methods may be affected. In this paper, an end-to-end feature learning model is proposed to extract denoising low-dimensional representations from distorted image datasets. These denoising features perform well in unsupervised clustering task. As the structural characteristics of this model, it can obtain essential information from image to decrease reconstruction error. Facing corrupted data, it also runs a better result. For evaluating their performance, these features are sent into k-means clustering [14]. Three evaluation metrics are selected for comparison including clustering accuracy (ACC), normalized mutual information (NMI) and adjusted rand index (ARI).

The following parts of this paper are arranged as follows. Some works related to classical feature learning are showed in Section 2. In Section 3, the structure and working mechanism of denoising feature learning model are presented. In Section 4, extensive experiments on eight standard datasets illustrate the effectiveness of presented model. Eventually, this paper is concluded in Section 5.

2. RELATED WORKS

In this section, we introduce several classical feature learning algorithms which are classified into two kinds: unsupervised feature learning and supervised feature learning.

2.1. Unsupervised Feature Learning

The unsupervised feature learning algorithms are categorized into two types: linear and non-linear. The core of linear feature learning algorithms is to obtain a linear mapping relation. Principal component analysis and isometric projection are commonly used linear feature learning algorithms. As for non-linear datasets, linear feature learning algorithms probably meet some

problems. Neighbourhood preserving embedding [15] and isometric feature mapping are two famous non-linear feature learning algorithms.

2.1.1. Isometric Projection

Isometric projection is a linear feature learning algorithm. Nevertheless, isometric projection can handle more complex datasets such as manifold data [16] which is embedded in high-dimensional space. Given a dataset $X \in \mathbb{R}d\times n$, isometric projection finds a mapping function f that makes $yi = f(xi)$ where $\{yi\ \}i=1\ n \in \mathbb{R}k$. Isometric projection defines dM the geodesic distance [17] measure on M which is a non-linear manifold embedded in $\mathbb{R}d$ and dE the standard Euclidean distance. Then optimization objective function is formalized as follows

$$\arg\min_{f} \sum_{i,j}(d_M(x_i,\ x_j) - d_E(f(x_i),\ f(x_j)))^2 \qquad (1)$$

where the mapping function f is to let Euclidean distances can offer an effective approximation to the geodesic distances on M.

2.1.2. Isometric Feature Mapping

Isometric feature mapping is a kind of manifold learning [18] method which is used in feature learning of non-linear data. Isomap algorithm has three steps. First step confirms neighbourhood for each point. There are two ways: k nearest neighbours (k-Isomap) and all points in radius ϵ (ϵIsomap). $d(xi\ ,\ xj)$ represents distance in input space, such that we obtain a weighted graph G. Second, if xi and xj are linked by an edge, initialize shortest path distances $d_G(x_i,\ x_j) = d(x_i,\ x_j)$ or else $d_G\ (x_i,\ x_j) = \infty$. Then $d_G(x_i,\ x_j)$ is constantly replaced by $\min\{d_G(x_i,\ x_j),\ d_G(x_i,\ x_p) + d_G(x_p,\ x_j)\}, p = 1, 2, \dots, N$. N is the number of whole points. Afterwards Isomap creates a matrix DG that consists of the shortest path distances. In the finally step, MDS is applied in DG . Consider the k -dimensional Euclidean space Y that preserves most information of manifolds intrinsic geometry, DY matrix is composed of Euclidean distances $\{d_Y(i,\ j) = \|y_i - y_j\|\}$. Then the cost function is denoted as

$$E = \|\gamma(DG\ ) - \gamma(DY)\|L\ 2 \qquad (2)$$

Where γ indicates an operator that converts distances to inner products.

2.1.3. Principal Component

Analysis Principal component analysis is a statistical method. It uses orthogonal transformation to convert a set of variables that may be related into a set of linearly uncorrelated variables. The converted set of variables is named the principal component. Consider an input image dataset $X = [x_1, \dots\ ,\ x_n]$, we obtain its normalized matrix X'. Covariance matrix C can be presented as follows

$$C = \frac{1}{n} X' X'^T \qquad (3)$$

Then we calculate the eigen values and eigenvectors about covariance matrix C. After that, k eigenvectors corresponding to k largest eigen values are selected. These eigenvectors are utilized to construct the projection matrix W which is ordered by eigen values descend. Finally, low dimensional feature representations $Y \in \mathbb{R}k\times n$ are formalized by: $Y = WX$.

2.2. Supervised Feature Learning

Supervised feature learning algorithms require sufficient labels, nevertheless, they perform a commendable result. In some cases, supervised feature learning algorithms can be improved into semi-supervised and then significantly reduce the need for labels. In this section, we mainly introduce supervised feature learning algorithms linear discriminant analysis, locality sensitive discriminant analysis and semi-supervised algorithm stacked label consistent auto-encoder (SLCA).

2.2.1. Linear Discriminant

Analysis As a classical algorithm in pattern recognition, the basic idea of linear discriminant analysis is to project high-dimensional pattern samples into the optimal discriminant vector space. Given a dataset$\{x_1, \dots, x_n\} \in \mathbb{R}^d$,$\{y_1, \dots, y_n\} \in \mathbb{R}^k$ $(k \ll d)$, attempt to find mapping matrix $A = (a_1, \dots, a_k) \in \mathbb{R}^{d \times k}$ such that $y_i = A^T x_i$. Suppose all samples are sorted into c classes. The objective function is denoted as follows

$$a_{opt} = \arg\max_a \frac{a^T S_b a}{a^T S_w a} \qquad (4)$$

$$S_b = \sum_{i=1}^{c} m_i (u_i - u)(u_i - u)^T \qquad (5)$$

$$S_w = \sum_{i=1}^{c} (\sum_{j=1}^{m_i} (x_j^i - u_i)(x_j^i - u_i)^T) \qquad (6)$$

where Sw is within-class scatter matrix while Sb is between-class scatter matrix. u means the total sample mean vector and mi is the number of data points in i-th class. ui represents the average vector of i-th class. The eigenvectors related to the largest eigen values constitute the basic functions of LDA:

$$S_b a = \lambda S_w a \qquad (7)$$

the aim of LDA is to preserve global class relationship between sample points. And as a classification, it is hoped that the coupling degree between classes is low and the degree of aggregation within classes is high.

2.2.2. Locality Sensitive Discriminant

Analysis Locality sensitive discriminant analysis is a popular data-analytic tool which can discover the local manifold structure. Local structure is more important if lacking of sufficient training samples. LSDA defines a projection by finding the local manifold structure and the projection maximizes the margin between sample points. Given n data points $\{x_1, \dots, x_n\} \in \mathbb{R}^d$, denote $N(x_i) = \{x_i^1, \dots, x_i^k\}$ the k nearest neighbors of x_i and $l(x_i)$ the class label of x_i.For each data point, $N(x_i)$ is divided into two subsets $N_w(x_i)$ and $N_b(x_i)$. $N_w(x_i)$ indicates the neighbours sharing the same label while $N_b(x_i)$ means the neighbours owning different labels

$$N_w(x_i) = \{x_i^j | l(x_i^j) = l(x_i), 1 \le j \le k\}$$

$$N_b(x_i) = \{x_i^j | l(x_i^j) \ne l(x_i), 1 \le j \le k\} \qquad (8)$$

It's obvious that $N_w(x_i) \cap N_b(x_i) = \emptyset$ and $N_w(x_i) \cup N_b(x_i) = N(x_i)$.Then the weight matrices are defined as $W_{w,ij} = 1$ if $x_i \in N_b(x_j)$ or $x_j \in N_b(x_i)$. Let y $= (y_1, \dots, y_m)^T$ be a map, theobjective functions are formalized as

$$\min_W \sum_{ij} (y_i - y_j)^2 W_{w,ij} \qquad (9)$$

$$\max_W \sum_{ij} (y_i - y_j)^2 W_{b,ij} \qquad (10)$$

The objective function (9) attempts to ensure that y_i and y_j are close while x_i and x_j are close and own same label.Maximizing (10) is to ensure that y_i and y_j are far apart if x_i and x_j are close and have different labels.

2.2.3. Stacked Label Consistent Auto-encoder

Stacked label consistent auto-encoder is a semi-supervised method which combines reconstruction and classification [19]. Its architecture is consist of two-layer stacked auto-encoder. Stacked label consistent auto-encoder aims to create a linear map between innermost layer and class labels which constitutes the class label consistency penalty. The optimization objective function is presented as

$$\min_{W_1, W_2, W_1', W_2', D} \|X - W_1' \emptyset(W_2' \emptyset(W_2 \emptyset(W_1 X)))\|_F^2 + \lambda \|L - D\emptyset(W_2 \emptyset(W_1 X))\|_F^2 \qquad (11)$$

Here, X is the input data matrix, D the linear map and L the class labels. Wi and $Wi\ '$ represent the weight between layers. Existing backpropagation techniques can't learn this architecture because there are two outputs. Stacked label consistent auto-encoder solves this problem by the Split Bregman technique. Formulation (11) requires all input samples have corresponding class labels. However, it is difficult to gain all labels and semi-supervision method is allowed. This leads to

$$\min_{W_1, W_2, W_1', W_2', D} \|X - W_1' \emptyset(W_2' \emptyset(W_2 \emptyset(W_1 X)))\|_F^2 + \lambda \|L - D\emptyset(W_2 \emptyset(W_1 X_S))\|_F^2 \qquad (12)$$

where the training data $X = [X_U | X_S]$and the subscripts denote unsupervised or supervised.

3. UNSUPERVISED DENOISING FEATURE LEARNING FOR DISTORTED IMAGE

Facing with distorted images, existing unsupervised feature learning methods may be not robust. For solving unsupervised clustering task of distorted images, an end-to-end feature learning model is presented to extract denoising low-dimensional representations. As the model is based on auto-encoder, next we introduce the structure of auto-encoder. Auto-encoder is a neural network which uses back propagation. Consider an input image $X \in \mathbb{R}^{m \times n}$. Auto-encoder aims to reconstruct a matrix X' which is similar to input data. In this process, auto-encoder network makes a hidden representation $Y \in \mathbb{R}^{d \times n}$ from X ($d \ll$ m).

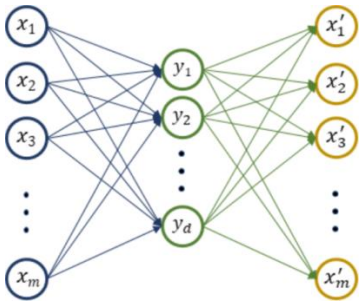

Figure 1. The structure of auto-encoder neural network

As demonstrated in Figure 1, the structure of auto-encoder is split into three parts: input layer, hidden layer and output layer. The process of encoder is denoted as $y = f(x)$and $x' = g(y)$means decoder. The optimization objective function of auto-encoder is represented as

$$\min_{W,b} \Theta(W,b) = \sum_{i=1}^{n} \|x_i - g(f(x_i))\|_2^2 \qquad (13)$$

where W and b mean the weight and the bias of neural network. Predefined activation function usually uses sigmoid function $S(x) = \frac{1}{1+e^{-x}}$.

If input datasets are distorted or suffer from noise, the features obtained by auto-encoder may be affected. Denoising feature learning model aims to enhance the robustness of feature. It is capable of reconstructing clean data from distorted data. Sometimes the reconstructed images could obtain a better performance than original images. Meanwhile it reduces the risk of over-fitting.

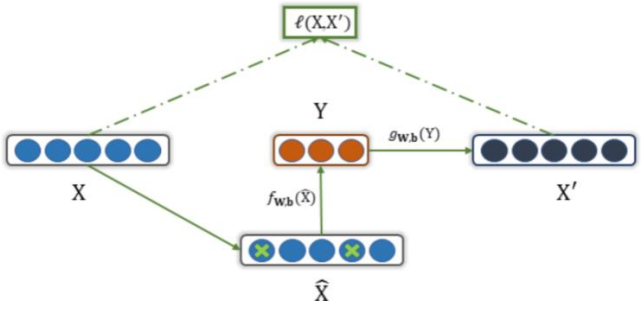

Figure 2. The framework of proposed model

The structure of proposed model is showed in Figure 2. Model accepts distorted data as input and output a clean data. Consider an original image dataset $X = [x_1, \dots, x_n] \in \mathbb{R}^{m \times n}$. As the influence of distortion or noise, corrupted dataset $\widehat{X}$ is formed. In the training process, corrupted dataset can besimulated by adding random zero into input data. Denoising low-dimensional representation is denoted as $Y \in \mathbb{R}^{d \times n}$.Finally, the reconstructed data X' can be presented as

$$Y = f_{W,b}(\hat{X}) = \alpha\left(W^{(1)}\hat{X} + b^{(1)}\right) \qquad (14)$$

$$X' = g_{W,b}(Y) = \alpha\left(W^{(2)}Y + b^{(2)}\right) = \alpha\left(W^{(2)}\alpha\left(W^{(1)}\hat{X} + b^{(1)}\right) + b^{(2)}\right) \qquad (15)$$

where α means activation function. Ordinarily activation function is using logistic sigmoid function $\alpha(x) = \frac{1}{1+e^{-x}}$. Each layer is defined to share the same network parameters. As a consequence, the weight $W^{(1)} = W^{(2)} = W$ and bias $b = [b^{(1)}; b^{(2)}]$. W is a$d \times m$ matrix and b is an-dimensional vector. The optimization objective function of model is presented as

$$\min_{W,b} \Theta(W, b) = L(X, X') = \sum_{i=1}^{n} L(x_i, x_i') = \sum_{i=1}^{n} \|x_i - g(f(\hat{x}_i))\|_2^2 \qquad (16)$$

where L is a cross-entropy loss function. The parameters of model network are denoted as $\theta = \{W, b\}$.They are constantly renovated via iterative descent of L. The detailed steps of denoising feature learning model are summarized as Algorithm 1.

4. EXPERIMENTAL ANALYSIS

In experiment stage, first we introduce eight public image databases. Then three popular clustering evaluation metrics and their working principles are demonstrated in the second section. Except presented model, seven common dimensional reduction algorithms are used to obtain feature representations. Experimental results on eight image datasets are recorded in three tables.

Algorithm 1 Algorithm for presented model

Input: Original data $\boldsymbol{X} = \{\boldsymbol{x_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{n}}$in $\mathbb{R}^{\boldsymbol{m}}$, the dimension of hidden layer $\boldsymbol{d}$ and learning rate $\boldsymbol{\sigma}$.
Output: Low-dimensional feature representation $\boldsymbol{Y} \in \mathbb{R}^{\boldsymbol{d \times n}}$.
1: Generate corrupted data $\widehat{\boldsymbol{X}} = \{\widehat{\boldsymbol{x}}_{\boldsymbol{i}}\}_{\boldsymbol{i=1}}^{\boldsymbol{n}}$;
2: Initialize weight matrix $\boldsymbol{W^{(1)}} \in \mathbb{R}^{\boldsymbol{d \times m}}$, bias vector $\boldsymbol{b^{(1)}}$and choose an activation function α.
3: **repeat**
4: **for**each point $\widehat{\boldsymbol{x}}_{\boldsymbol{i}}(\boldsymbol{i} = \boldsymbol{1}, \dots, \boldsymbol{n})$**do**
5: Compute $\boldsymbol{y_i}$by Formula (14);
6: Utilize Formula (15) to obtain $\boldsymbol{x_i'}$;
7: Update $\boldsymbol{W}$and $\boldsymbol{b}$by the following Formula $\boldsymbol{W^{(i+1)}} \leftarrow \boldsymbol{W^{(i)}} - \boldsymbol{\alpha}\frac{\boldsymbol{\partial}}{\boldsymbol{\partial W^{(i)}}}\boldsymbol{\Theta(\theta)}$and $\boldsymbol{b^{(i+1)}} \leftarrow \boldsymbol{b^{(i)}} - \boldsymbol{\alpha}\frac{\boldsymbol{\partial}}{\boldsymbol{\partial b^{(i)}}}\boldsymbol{\Theta(\theta)}$with gradient descent method;
8: **end for**
9:**until** convergence
10:**return**$\boldsymbol{Y}$.

4.1. Data Sets

In this section, we will introduce eight public standard datasets. These image datasets are Chars74K, USPS, Yale-B, COIL-20, ORL, CIFAR-10, Fashion-MNIST, SMSHP. The details of them are given in Table I. Specific description of eight image datasets are showed below.

The Chars74K dataset [20] contains two parts: English and Kannada. English symbols have three kinds. First kind contains 7705 characters come from natural images. Second one has 3410 hand drawn characters which use a tablet PC. The last one has 62992 synthesised characters which originate from computer fonts. This dataset is divided into 62 classes (a-z, A-Z, 0-9) and the pixel

size of each image is 32*32. We select a subset of Chars74K dataset. It has 44,044 training images and 8788 test images with 52 classes (a-z, A-Z).

USPS is a handwritten digit image dataset [21]. It owns 9298 handwritten digit images in total. Size of each image is 16*16. USPS is divided into two parts: 7291 training samples and 2007 test samples. The two subsets contain 10 different categories. Label '1' means digit 1 and label '0' represents digit 10.

The extended Yale Face Database B (YaleB) [22] is a face image database. YaleB includes 38 individuals and each individual has 64 images. We resize these image into 32*32 pixels. YaleB is divided to two subsets. Training one has 1928 samples and test one has 486 samples. They contain 38 different classes.

Columbia University Image Library (COIL-20) is an object image dataset. It is gray-scale. COIL20 has 20 objects and each object owns 72 images. They are taken from different angles. The size of these image is 32*32 pixels. COIL-20 contains 1440 samples. Each sample is represented by a 1024-dimensional vector. We divide the dataset into two subsets. First has 1140 training examples and second owns 300 test examples.

Olivetti Research Laboratory (ORL) [23] is a face image dataset. It contains 40 subjects with different ages, sexes and races. There are 10 images in each subject. ORL was made at different times, varying the lighting, facial details (glasses / no glasses) and expressions (smiling / not smiling, open / closed eyes). Each image is resized to 1024-dimensional vector. The dataset has 40 classes in all.

Cifar-10 is a standard color image dataset. It is made up of 60000 images which originate from a larger scale dataset. Cifar-10 contains 10 classes (cat, dog, automobile, bird, airplane, deer, ship, frog, horse, truck). There are 6000 images in each class. The size of image is 32*32. It is split into two subsets. Training samples have 1928 images and test samples own 486 images. They contain 38 different classes.

Fashion-MNIST is a clothing image dataset. It contains 10 classes (bag, coat, trouser, shirt, sandal, T-shirt, dress, pullover, sneaker, ankle boot). Fashion-MNIST includes 60,000 training samples and 10,000 testing samples. The size of each image is 28*28 pixels. Each sample is represented as a 784-dimensional feature vector.

SMSHP (Sebastien Marcel Static Hand Posture) is a hand-posture image dataset [24]. It consists of 5531 images. SMSHP is divided into 6 different types (point, five, v, a, b, c). For simplicity, the size of these hand posture images is denoted as 32*32 pixels. They are split into two subsets. First one has training images and second one owns 1106 test images. Each example is unified as a 1,024-dimensional feature vector.

Table 1. A brief description of the tested datasets.

ID	datasets	# samples	# features	# classes
1	Chars74K	52832	1024	52
2	COIL-20	1440	1024	20
3	USPS	9298	256	10
4	ORL	400	1024	40
5	YaleB	2414	1024	38
6	Cifar-10	60000	3072	10
7	SMSHP	5531	1024	6
8	Fashion-MNIST	70000	784	10

4.2. Parameter Setting

In this paper, learning rate of all methods is set as 0.01. For an objective comparison, we reduce each dataset into k-dimension uniformly. Where the number of hidden layer units k is set as 40. The size of each bach is denoted as 100. And we fix the number of training as 50. For simulating distorted image, we add random noises into those eight image datasets. Ten samples from each processed dataset are demonstrated in Figure 3.

Figure 3. The samples of distorted datasets.

4.3. Evaluation Metrics

In this section, we mainly introduce the evaluation metric of clustering. In the final stage of experiment, the k-means algorithm is used to calculate performance of extracted features. Consider a sample dataset $D = \{x_1, \dots, x_n\}$.The clusters obtained by k-means algorithm for clustering are denoted as $C = \{C_1, \dots, C_n\}$.Then the square error can be computed by

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - u_i\|_2^2 \qquad (17)$$

where $u_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$.The core of algorithm is to minimize E.

Clustering accuracy is an important reference index of clustering performance [25]. It is used to compare predicted labels with real labels provided by data. The value of clustering accuracy can be presented as

$$ACC = \frac{\sum_{i=1}^{n} \delta(s_i, map(r_i))}{n} \qquad (18)$$

where r_i and s_i represent predicted label and real label separately. The number of data is set as n. $\delta(x, y) = 1$ if $x = y$, otherwise$\delta(x, y) = 0$.Normalized mutual information can be used to measure the similarity of clustering results [26]. Consider a mutual information $I = (\Omega; C)$.It represents the increase of category information Ω by giving cluster information C. Then normalized mutual information is presented as

$$NMI = \frac{I(\Omega; C)}{(H(\Omega) + H(C))/2} \qquad (19)$$

where H means entropy. Adjusted Rand index is a functionto calculate the distribution similarity of two labels [27]. Thisfunction has no requirements for the definition form of label.

$$ARI = \frac{\sum_{ij} \binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}} \qquad (20)$$

where $ARI \in [-1, 1]$.The higher the value, the more consistent the clustering results with the real situation.

4.4. Experimental Results

In this section, comprehensive experiments are presented. In addition to denoising feature learning model, we run several classical feature learning algorithms for comparison. These methods include PCA, NPE, LPP, Isomap, LLE, IsoP and auto-encoder. For assessing the performance of feature representation, we choose k-means algorithm to make a clustering. Three popular evaluation metrics ACC, NMI and ARI are used for revealing an intuitive result. Meanwhile the original data without dimensionality reduction is also sent into k-means as the baseline. Finally, all the experimental results on eight processed image datasets are displayed in three tables.

From the Table 2 - 3, it is intuitive that the low-dimensional feature representations extracted by denoising model run a better performance. In Table 2, the clustering accuracy of denoising model ranks first on seven image datasets except USPS. In Yale-B dataset, the feature representations extracted via denoising model perform a favorable result compared with the baseline. In Table 3, the performance of denoising model reach first on six datasets. On Chars74K image dataset, normalized mutual information of denoising model is 60.8% while baseline is 45.3%, with a greater improvement. On CIFAR-10, denoising model reaches second best result which is close to locality preserving projections. In Table 4, the adjusted rand index of denoising model ranks first on six datasets while classic unsupervised feature learning algorithms also perform well. Especially on the extended Yale Face Database, the adjusted rand index of denoising model reaches a bigger improvement.

Table 2. Clustering accuracy (mean% + std%) with different unsupervised feature learning methods

dataset/method	Chars74K	USPS	Yale-B	COIL-20	ORL	CIFAR-10	F-MNIST	SMSHP
Baseline	31.8±0.9	65.6±1.8	11.3±0.4	56.1±2.7	63.6±1.8	23.9±0.3	54.4±1.2	38.7±1.5
PCA	34.5±1.6	68.2±2.3	12.2±0.3	65.7±1.4	64.8±2.6	24.5±1.2	59.3±0.7	36.5±1.1
NPE	35.6±0.8	**71.3±1.8**	28.2±1.5	61.3±2.3	72.7±1.4	21.3±0.5	52.4±0.9	36.8±1.2
LPP	33.2±1.5	68.5±1.4	30.3±1.4	66.7±0.5	68.6±2.5	24.2±0.8	58.1±3.3	38.3±0.9
Isomap	24.6±0.3	67.3±2.6	32.6±2.1	68.6±1.7	59.8±2.6	23.9±1.2	57.2±2.9	34.5±0.4
LLE	28.9±1.2	64.2±1.5	27.4±1.7	60.2±1.1	53.6±1.7	25.4±2.3	53.7±1.8	33.6±1.5
IsoP	34.3±2.5	70.2±0.9	25.3±0.8	65.3±2.8	62.4±2.1	26.3±1.6	54.2±1.3	35.2±1.2
Auto-encoder	32.6±1.2	67.4±2.4	19.7±0.4	59.9±1.3	65.6±3.5	29.7±0.9	58.2±2.6	34.3±0.8
Ours	**37.2±0.6**	69.5±0.8	**33.8±1.2**	**70.4±2.2**	**74.5±2.3**	**32.5±1.1**	**61.5±1.4**	**39.9±0.6**

Table 3. Normalized mutual information (mean% + std%) with different unsupervised feature learning methods.

dataset/method	Chars74K	USPS	Yale-B	COIL-20	ORL	CIFAR-10	F-MNIST	SMSHP
Baseline	45.3±0.7	63.6±1.6	12.8±0.5	74.9±1.8	73.4±2.2	8.6±0.6	54.5±1.4	7.1±0.8
PCA	50.6±2.3	61.0±0.2	14.3±0.4	76.7±2.3	76.8±1.5	8.2±0.4	53.2±1.6	8.4±0.6
NPE	55.5±0.8	63.2±0.8	37.6±1.6	74.6±0.6	80.2±3.2	8.5±0.3	52.1±0.7	11.9±0.8
LPP	53.4±0.9	67.6±1.9	35.4±0.3	76.8±1.4	77.9±1.8	**9.8±0.6**	58.7±2.2	9.3±1.2
Isomap	45.3±1.2	65.9±2.2	36.8±2.0	73.5±2.1	74.1±2.3	9.2±0.1	54.6±0.8	8.8±0.7
LLE	52.7±1.6	63.2±2.8	23.5±1.4	72.3±4.2	73.8±2.7	7.6±0.2	53.4±0.6	**12.0±0.5**
IsoP	49.6±2.3	58.3±1.5	29.3±2.3	78.5±3.6	75.7±1.3	8.2±0.7	52.6±0.9	9.2±0.3
Auto-encoder	49.2±1.5	56.7±2.4	22.6±1.2	75.2±1.9	77.3±2.6	7.4±0.3	56.2±1.8	8.3±0.5
Ours	**60.8±2.2**	**68.1±2.3**	**39.6±0.8**	**79.1±2.3**	**82.4±2.4**	9.6±0.2	**62.5±1.3**	10.1±0.6

Table 4. Adjusted rand index (mean% + std%) with different unsupervised feature learning methods.

dataset/method	Chars74K	USPS	Yale-B	COIL-20	ORL	CIFAR-10	F-MNIST	SMSHP
Baseline	20.8±0.9	53.6±2.2	2.3±0.2	50.7±3.6	48.4±2.1	4.5±0.6	38.6±0.9	4.4±0.2
PCA	23.5±0.4	60.2±0.6	3.5±0.4	55.3±1.8	49.6±2.0	5.1±0.6	41.2±0.7	6.1±0.3
NPE	24.3±2.2	56.6±0.7	13.6±0.3	53.3±2.5	53.6±1.4	5.7±0.4	33.7±1.3	5.4±0.1
LPP	20.7±0.8	62.2±1.3	12.4±0.5	61.8±3.2	46.8±2.3	4.2±0.1	42.3±2.5	4.8±0.9
Isomap	21.1±0.6	62.7±2.4	13.8±0.2	60.5±1.3	38.9±0.6	4.7±0.6	40.7±1.8	3.7±0.2
LLE	19.3±0.4	56.7±3.2	9.2±0.8	51.4±2.6	45.6±0.7	6.1±0.2	**46.3±0.7**	4.5±0.3
IsoP	22.5±0.3	58.2±2.8	11.8±0.7	**64.3±3.2**	40.4±1.5	6.7±0.5	43.6±1.2	5.1±0.2
Auto-encoder	23.2±0.6	61.8±3.5	6.9±0.2	52.5±2.4	42.3±2.3	7.3±0.8	42.1±0.5	7.3±0.6
Ours	**26.4±0.5**	**65.9±2.3**	**14.6±0.4**	54.9±2.1	**55.7±1.6**	**7.5±0.4**	44.7±0.3	**8.8±0.4**

5. CONCLUSION

In this paper, facing the problem regard to high-dimensional of distorted images, an end-to-end denoising feature learning model was proposed to obtain high robust feature representations. Then the extracted features were evaluated by *k*-means clustering. Compared to other unsupervised feature learning methods, extensive experiments on eight processed image datasets demonstrated that denoising model ran a competitive performance. The low-dimensional representation could replace the original dataset primely. But in the experiment, it was obvious that larger dimensions and categories caused a bad influence on clustering performance. In the future work, we will be concerned with the image datasets which own many categories. We may add semi-supervised training to attempt a better result.

ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China under Grants No. 61877010 and No. 11501114, and the Fujian Natural Science Funds under Grant No. 2019J01243. Genggeng Liu is the corresponding author of the article.

REFERENCES

[1] S. Wang, W. Zhu, Sparse graph embedding unsupervised feature selection, IEEE Transactions on Systems, Man, and Cybernetics: Systems, 48 (3), pp. 329–341, 2018.
[2] S. Wang, W. Pedrycz, Q. Zhu, W. Zhu, Subspace learning for unsupervised feature selection via matrix factorization, Pattern Recognition, 48 (1), pp. 10–19, 2015.
[3] S. Wang, J. Lu, X. Gu, H. Du, J. Yang, Semi-supervised linear discriminant analysis for dimension reduction and classification, Pattern Recognition, 57 (C), pp. 179–189, 2016.
[4] D. Cai, X. He, K. Zhou, J. Han, H. Bao, Locality sensitive discriminant analysis, In Proceedings of the 20th International Joint Conference on Artifical Intelligence, pp. 708–713, 2007.
[5] H. Hotelling, Analysis of a complex of statistical variables into principal components, Journal of Educational Psychology, 24 (6), pp. 417–520, 1933.
[6] X. He, Locality preserving projections, Advances in Neural Information Processing Systems, 16 (1), pp. 186–197, 2003.
[7] Z. Huang, X. Xu, L. Zuo, Reinforcement learning with automatic basis construction based on isometric feature mapping, Information Sciences, 286, pp. 209–227, 2014.
[8] S. T. Roweis, L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science, 290 (5500), pp. 2323–2326, 2000.
[9] C. Deng, X. He, J. Han, Isometric projection, in: National Conference on Artificial Intelligence, pp. 528–533, 2007.
[10] Y. Wang, H. Yao, S. Zhao, Auto-encoder based dimensionality reduction, Neurocomputing, 184 (C), pp. 232–242, 2016.

[11] J. Deng, Z. Zhang, E. Marchi, B. Schuller, Sparse autoencoder-based feature transfer learning for speech emotion recognition, in: Affective Computing and Intelligent Interaction, pp. 511–516, 2013.
[12] J. Walker, C. Doersch, A. Gupta, M. Hebert, An uncertain future: forecasting from static images using variational autoencoders, in: European Conference on Computer Vision, Springer, pp. 835–851, 2016.
[13] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434.
[14] P. Fränti, S. Sieranoja, K-means properties on six clustering benchmark datasets, Applied Intelligence, 48 (12), pp. 4743–4759, 2018.
[15] X. He, D. Cai, S. Yan, H. J. Zhang, Neighborhood preserving embedding, in: Tenth IEEE International Conference on Computer Vision, pp. 1208–1213, 2005.
[16] A. N. Gorban, B. Kgl, D. C. Wunsch, A. Y. Zinovyev, Principal manifolds for data visualization and dimension reduction, Springer Berlin Heidelberg, 2008.
[17] C. Varini, A. Degenhard, T. W. Nattkemper, Isolle: lle with geodesic distance, Neurocomputing, 69 (13), pp. 1768–1771, 2006.
[18] B. Raytchev, I. Yoda, K. Sakaue, Head pose estimation by nonlinear manifold learning, in: IEEE Proceedings of the 17th International Conference on Pattern Recognition, Vol. 4, pp. 462–466, 2004.
[19] A. Gogna, A. Majumdar, R. Ward, Semi-supervised stacked label consistent autoencoder for reconstruction and analysis of biomedical signals, IEEE Transactions on Biomedical Engineering, 64 (9), pp. 2196–2205, 2016.
[20] C. Yi, X. Yang, Y. Tian, Feature representations for scene text character recognition: A comparative study, in: 12th International Conference on Document Analysis and Recognition, IEEE, pp. 907–911, 2013.
[21] K. Proedrou, I. Nouretdinov, V. Vovk, A. Gammerman, Transductive confidence machines for pattern recognition, in: European Conference on Machine Learning, Springer, pp. 381–390, 2002.
[22] A. S. Georghiades, P. N. Belhumeur, D. J. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Transactions on Pattern Analysis and Machine Intelligence, 23 (6), pp. 643–660, 2001.
[23] G. Guo, S. Z. Li, K. Chan, Face recognition by support vector machines, in: Fourth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 196–201, 2000.
[24] W. Guo, G. Chen, Human action recognition via multi-task learning base on spatial–temporal feature, Information Sciences, 320, pp. 418–428, 2015.
[25] K. A. Nazeer, M. Sebastian, Improving the accuracy and efficiency of the k-means clustering algorithm, in: Proceedings of the World Congress on Engineering, Vol. 1, pp. 1–3, 2009.
[26] Z. F. Knops, J. A. Maintz, M. A. Viergever, J. P. Pluim, Normalized mutual information based registration using k-means clustering and shading correction, Medical Image Analysis, 10 (3), pp. 432–439, 2006.
[27] J. M. Santos, M. Embrechts, On the use of the adjusted rand index as a metric for evaluating supervised classification, in: International Conference on Artificial Neural Networks, Springer, pp. 175–184, 2009.

AUTHORS

Qihao Lin

Qihao Lin received the B.E. degree in Network Engineering from Fuzhou University, Fuzhou, China, in 2018. He is currently pursuing the M.S. degree with the College of Mathematics and Computer Science, Fuzhou University. His research interests include machine learning and computer vision.

Jinyu Cai

Jinyu Cai received the B.S. degree in computer science and technology from Fuzhou University, Fuzhou, China, in 2018. He is currently pursuing the M.S. degree with the College of Mathematics and Computer Science, Fuzhou University. His research interests include machine learning, computer vision and pattern recognition.

Genggeng Liu

Genggeng Liu received the B.S. degree in computer science and the Ph.D. degree in applied mathematics from Fuzhou University, Fuzhou, China, in 2009 and 2015, respectively. He is currently an Associate Professor with the College of Mathematics and Computer Science, Fuzhou University. His contributions have been published in IEEE Transactions on Cybernetics, IEEE Transactions on Industrial Informatics, ACM Transactions on Design Automation of Electronic Systems, etc. His research interests include computational intelligence and its application.