DISTRIBUTION PRESERVING NETWORK EMBEDDING

Anyong Qin, Zhaowei Shang, Taiping Zhang*

Chongqing University, China

Yuan Yan Tang

University of Macau, Macau, China

ABSTRACT

The deep autoencoder network which is based on constraining non-negative weights, can learn a low dimensional part-based representation. On the other hand, the inherent structure of the each data cluster can be described by the distribution of the intraclass sample. Then one hopes to learn a new low dimensional feature which can preserve the intrinsic structure embedded in the high dimensional data space perfectly. In this paper, by preserving data distribution, a deep part-based representation can be learned, and the novel algorithm is called Distribution Preserving Network Embedding (DPNE). In DPNE, we first need to estimate the distribution of the original data, and then we seek a part-based representation which respects the distribution. The experimental results on real-world data sets show that the proposed algorithm has good performance in terms of cluster accuracy and adjusted mutual information (AMI).

***Index Terms*—** Distribution preserving, manifold structure, part-based representation, sparse autoencoder, clustering.

1. INTRODUCTION

Unsupervised learning is a branch of machine learning algorithm that can learn inherent information from the unlabeled observation data. In many problems, the observation sample matrix is of very high dimension, for example, image and document recognition. So it is infeasible to process the original data directly [1]. One hopes then to learn a lower dimensional representation which still retains the inherent structure of the original data. In the last decade, a large number of linear and nonlinear methods for unsupervised dimensionality reduction have been proposed [2, 3].

The deep learning method, multilayer autoencoders are the stacked feed-forward neural network and attempt to learn a complex nonlinear mapping function, which can automatically learn interesting feature from the input data [4]. For this reason, the deep autoencoder has been applied to various pattern recognition tasks, such as image and document [5–7]. Recently, the novel clustering methods, Deep Embedding Clustering (DEC) [6] and Deep Clustering Network (DCN) [7] both make full use of the deep autoencoder to learn the latent representation, and achieve good performance. On the other hand, by constraining the non-negative property onto the two decomposition factors, NMF has been shown to be suitable for learning the parts of objects [8]. So the non-negative constraint autoencoder (NCAE) method is proposed to learn a part-based representation using deep autoencoder with non-negative constraint [9].

However, these algorithms do not well capture or even ignore the nonlinear manifold structure imbedded in the high dimensional data space. In general, the data is sampling from the probability distribution that are near to a submanifold of the ambient space [10, 11]. Moreover, the nonlinear manifold structure of each class can be depicted by the distribution of the intraclass members. The intraclass samples are general located in a continuous high density area called density connected, while the different cluster are connected by some low density area called the border [12, 13]. Preserving this manifold structure in the low dimensional feature space contributes to finding a clean and discriminative representation [14].

In this paper, we demonstrate how to learn a meaningful data representation that explicitly considers the manifold structure. And we propose a novel dimensionality reduction technique, called Distribution Preserving Network Embedding (DPNE), which can learn a part-based representation using NCAE and simultaneously respect the distribution of high dimensional data space. By estimating the distribution of high dimensional space, we can encode the manifold information embedded in the high dimensional data space. The goal of the proposed dimensionality reduction technique is to learn a meaningful data representation that can preserve the distribution of the high dimensional data space as much as possible. As a consequence, two points that locate in a high density area are putting as close together as possible in the low dimensional feature space, and two points that are connected by low density area are far apart in the low dimensional representation.

2. RELATED WORK

An autoencoder neural network is one unsupervised learning algorithm, which can automatically learn feature and then reconstruct its input at the output layer [15]. It tries to learn two functions, i.e., encoder function $F(x)$ and decoder function

*Corresponding author (e-mail: ayqin@cqu.edu.cn). The work was supported by the National Natural Science Foundation of China (61672114).

$G(F(x))$. The encoder function $F(x)$ maps the input data to the feature space. Specifically, the computation of hidden representation is given by

$$h^{(1)} = F(x) = \sigma(\omega^{(1)}x + b^{(1)}) \tag{1}$$

where x is the input data, $\omega^{(1)}$ denotes the weight, $b^{(1)}$ represents the bias, and $\sigma(\cdot)$ is the activation function. The decoder function $G(F(x))$ reconstructs the input data according to the representation space. And the computation of reconstructing the input data is as follows,

$$\hat{x} = G(h^{(1)}) = \sigma(\omega^{(2)}h^{(1)} + b^{(2)}) \tag{2}$$

where $\omega^{(2)}$ denotes the weight and $b^{(2)}$ represents the bias. To optimize the parameters of the autoencoder neural network, i.e., $\Omega = \{\omega^{(1)}, \omega^{(2)}, b^{(1)}, b^{(2)}\}$, the average reconstruction error is used as the objective function,

$$\mathcal{O}(\Omega) = \min_{\Omega} \frac{1}{N}\sum_{i=1}^{N} \|x_i - \hat{x}_i\|_F^2 \tag{3}$$

where N represents the number of input data.

Using the stack autoencoder contributes to discovering the latent structure embedded in the high dimensional space [4, 16]. Generally, a weight decay term is added to the (3) to help prevent overfitting [17]. And the final cost function of autoencoder is defined as

$$\mathcal{O}(\Omega) = \frac{1}{N}\sum_{i=1}^{N} \left\|x_i - G(h_i^{(\frac{L}{2})})\right\|_F^2 + \frac{\beta}{2}\sum_{l=1}^{L}\sum_{i=1}^{s_{l-1}}\sum_{j=1}^{s_l}(\omega_{ij}^{(l)})^2 \tag{4}$$

where $\Omega = \{\omega^{(l)}, b^{(l)}\}$ denotes the parameters of the model with multiple hidden layers, the even L denotes the number of layers, β denotes the regularization parameter, s_{l-1} and s_l are the sizes of adjacent layers, and $h_i^{(\frac{L}{2})}$ denotes the output of $\frac{L}{2}$-th layer, i.e., the final low dimensional representation of sample x_i.

When the number of hidden units is large, imposing some constraints on the hidden layers contributes to maintaining the proper number of active neuron [17]. Imposing a sparsity constraint on the hidden units, the autoencoder will still extract the latent structure hidden in the input data [18]. Enforcing the activation of hidden units to be near 0 is a common imposing the sparsity constraint method [19]. The average activation of the hidden unit j is defined as

$$\hat{p}_j = \frac{1}{N}\sum_{i=1}^{N} H_{ij} \tag{5}$$

where H_{ij} denotes the j-th element of the i-th hidden representation (i.e., h_i). By constraining the $\hat{p}_j = p$ (p is a small positive value close to 0, for example, $p = 0.05$), the sparsity can be enforced [17]. Using the Kullback-Leibler divergence can achieve the constraint, i.e., $\hat{p}_j = p$

$$KL(p \parallel \hat{p}) = \sum_{j=1}^{s_l} p\, log\frac{p}{\hat{p}_j} + (1-p) log\frac{1-p}{1-\hat{p}_j} \tag{6}$$

To achieve sparsity constraint for the autoencoder neural network, an extra penalty term is added to the objective function (4), and the final objective function of sparse autoencoder (SAE) can be written as

$$\begin{aligned}\mathcal{O}(\Omega) = \quad & \frac{1}{N}\sum_{i=1}^{N} \left\|x_i - G(h_i^{(\frac{L}{2})})\right\|_F^2 + \alpha KL(p \parallel \hat{p}) \\ & + \frac{\beta}{2}\sum_{l=1}^{L}\sum_{i=1}^{s_{l-1}}\sum_{j=1}^{s_l}(\omega_{ij}^{(l)})^2\end{aligned} \tag{7}$$

Many researches have demonstrated that utilizing the benefits of part-based representation with non-negative constraint can improve the performance of deep neural network [20, 21]. To encourage the connecting weight $\omega^{(l)}$ to be non-negative, the regularization parameter in (7) is replaced as a quadratic function [20]. Thus, the objective function of non-negative constraint autoencoder (NCAE) can be written as

$$\begin{aligned}\mathcal{O}(\Omega) = \quad & \frac{1}{N}\sum_{i=1}^{N} \left\|x_i - G(h_i^{(\frac{L}{2})})\right\|_F^2 + \alpha KL(p \parallel \hat{p}) \\ & + \frac{\beta}{2}\sum_{l=1}^{L}\sum_{i=1}^{s_{l-1}}\sum_{j=1}^{s_l} J(\omega_{ij}^{(l)})\end{aligned} \tag{8}$$

where

$$J(\omega_{ij}^{(l)}) = \begin{cases}(\omega_{ij}^{(l)})^2, & \omega_{ij}^{(l)} < 0, \\ 0, & \omega_{ij}^{(l)} > 0.\end{cases} \tag{9}$$

3. THE PROPOSED APPROACH

3.1. The Standard Kernel Density Estimation

Our early researches in manifold learning [13, 14] have demonstrated that preserving the nonlinear manifold structure of original data in the low dimensional space can be achieved by minimizing the inconsistence of two distributions [22]. Due to the lack of prior knowledge, it is desirable to use the popular kernel density estimation to approximate the truthful distributions of data in high dimensional space and low dimensional space, respectively.

$$f(x) = \sum_{i=1}^{N} f(x|x_i) = \frac{c}{Nb_x^M}\sum_{i=1}^{N}\kappa\left(\left\|\frac{x - x_i}{b_x}\right\|^2\right) \tag{10}$$

$$g(h) = \sum_{i=1}^{N} g(h|h_i) = \frac{c}{Nb_h^D}\sum_{i=1}^{N}\kappa\left(\left\|\frac{h - h_i}{b_h}\right\|^2\right) \tag{11}$$

where $\kappa(\|x\|^2)$ is called $profile$ of the kernel, M is the dimension of the original high dimensional data space, D is the dimension of the learned representation h_i and c is a positive normalization constant [23, 24]. The parameters b_x and b_h should satisfy the nonlinear equation $\sum_{i=1}^{N} \mathcal{K}_B(x - x_i) log_2(\mathcal{K}_B(x - x_i)) = log_2(t)$ and t is set to 20 [13]. To preserve the distribution of original data in the low dimensional feature space, Kullback-Leibler divergence criterion can be used to measure the inconsistency,

$$f(x) \log \frac{f(x)}{g(h)} \tag{12}$$

3.2. k-Nearest Neighbour Kernel Density Estimation

We use the kernel density estimation to capture the manifold structure. Thus, the key problem for the proposed method is how to estimate the density of samples. However, the standard kernel density estimation has poor performance in the high dimensional space. The k-nearest neighbor kernel density estimation is a special case of the standard kernel density estimation with the local variables. According to the number of samples in a local region, we can smooth this estimator to obtain the more approximate density [25, 26]. The k-nearest neighbor kernel density estimation described in [25, 26] can be written as

$$f(x, k) = \frac{c}{N d_x^M} \sum_{i=1}^{N} \mathcal{K}\left(\frac{x - x_i}{d_x}\right) \tag{13}$$

where $d_x = d(x)$ is a Euclidean distance between x and the k-th nearest neighbor of x among x_j's,

$$d(x) = min(k, \{|x - x_j|, j = 1, 2, \cdots, N\}), \tag{14}$$

where $min(k, A)$ is the k-th smallest element of the set A. So the density of the given data is rewritten as

$$f(x) = \sum_{i=1}^{N} f(x|x_i) = \frac{c}{N d_x^M} \sum_{i=1}^{N} \kappa\left(\left\|\frac{x - x_i}{d_x}\right\|^2\right) \tag{15}$$

3.3. DPNE

If we assume the learning low dimensional feature preserves the distribution of the given data very well, it follows that the conditional density of the given data at point x_i ($x_i \in X$) and the conditional density of the corresponding low dimensional representation at point h_i ($h_i \in H$) will be equal. In other words, the goal of distribution preserving is to find a lower dimensional representation H that minimizes the inconsistency between $f(x_i|x_j)$ and $g(h_i|h_j)$ for all (x_i, h_i). To achieve the goal, we try to minimize the inconsistency between any $f(x_i|x_j)$ and $g(h_i|h_j)$,

$$\sum_{i=1}^{N} \sum_{j=1}^{N} f(x_i|x_j) \log \frac{f(x_i|x_j)}{g(h_i|h_j)} \tag{16}$$

This paper does not focus on the choice of kernel function $\kappa(\cdot)$ in the step of kernel density estimation. So we employ the Gaussian kernel and Cauchy kernel to estimate the distributions of high dimensional space and low dimensional space, respectively. The Cauchy kernel is a long-tailed kernel and has the ability to alleviate the crowding problem in the low dimensional space [13, 14]. By minimizing (16), we expect that if two samples x_i and x_j are close, the low dimensional representations h_i and h_j are also close to each other, and vice versa. Thus, as an extra penalty term, we add (16) to the (8). The greedy layer-wise trained NCAE model is used to initialize the proposed DPNE network, and the resulting network is not imposed sparsity constraint in the fine-tuning stage. The final cost function of the proposed DPNE network is as follows

$$\begin{aligned} \mathcal{O}(\Omega, h) = & \frac{1}{N} \sum_{i=1}^{N} \left\| x_i - G(h_i^{(\frac{L}{2})}) \right\|_F^2 \\ & + \frac{\beta}{2} \sum_{l=1}^{L} \sum_{i=1}^{s_{l-1}} \sum_{j=1}^{s_l} J(\omega_{ij}^{(l)}) \\ & + \gamma \sum_{i=1}^{N} \sum_{j=1}^{N} f(x_i|x_j) \log \frac{f(x_i|x_j)}{g(h_i^{(\frac{L}{2})}|h_j^{(\frac{L}{2})})} \end{aligned} \tag{17}$$

where γ controls the penalty term facilitating distribution preserving. We use the backpropagation algorithm to compute the gradient descent of the (17) to optimize the network parameter Ω.

4. EXPERIMENTS

In this paper, we compare the representation space obtained by the proposed DPNE to the classic and deep model algorithms, i.e, k-means++ [27], Deep Embedding Clustering (DEC) [6], Deep Clustering Network (DCN) [7], Sparse Autoencoder (SAE) [17], and Non-negative Constraint Autoencoder (NCAE) [9]. For the SAE, NCAE and DPNE model, we use the k-means method to cluster the low dimensional representation in our paper. The cluster accuracy (ACC) and the adjusted mutual information (AMI) are employed to evaluate the performance of these different algorithms. The default parameters for the compared algorithms are used. The input parameters of our algorithm are as follows: regularization parameters $\beta = 0.003$ and $\gamma = 100$, learning rate $\eta = 0.1$, the number of layers $L = 8$, the number of iterations $maxiter = 400$ and the number of nearest neighbors $k = 10$. Same as the DEC and DCN, the list of the layer size is $M - 500 - 500 - 2000 - D$ ($M \gg D$) [6, 7]. For all methods, we repeat ten times to obtain reliable and stable results of each data set. We evaluated our proposed method against five widely used data sets: MNIST, Coil-100, YaleB, Reuters21578 and RCV1.

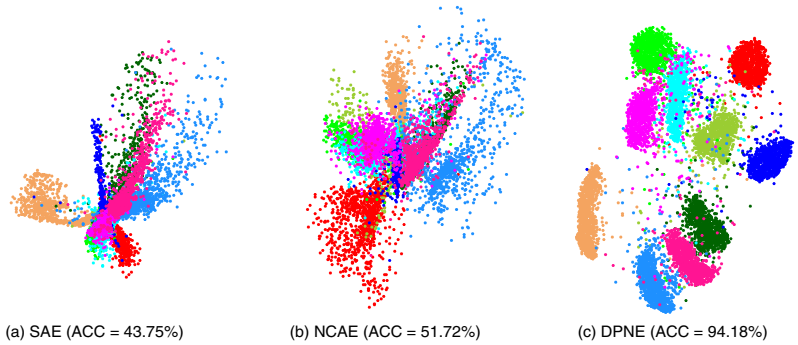

Fig. 1. Visualize the 2-dimensional representation space obtained by (a) SAE, (b) NCAE and (c) DPNE.

Table 1. Clustering results (%) of different methods.

	ACC					AMI				
	MNIST	Coil100	YaleB	Reuters	RCV1	MNIST	Coil100	YaleB	Reuters	RCV1
k-means++	53.2	62.1	51.4	23.6	52.9	50.0	80.3	61.5	51.6	35.5
DEC	86.7	81.5	2.7	16.8	68.3	84.0	61.1	0.0	39.7	**50.0**
DCN	56.0	62.0	43.0	22.0	**73.0**	57.0	81.0	59.0	43.0	47.0
SAE	60.9	54.8	47.5	23.7	49.1	56.3	32.5	62.7	48.3	43.8
NCAE	65.6	57.9	50.9	31.2	50.3	59.8	34.8	65.1	47.5	44.9
DPNE	**96.1**	**85.0**	**94.8**	**57.3**	56.2	**90.5**	**95.8**	**98.5**	**56.7**	**50.0**

4.1. Visualization

In this section, when the high dimensional data is embedded into a 2-dimensional plane, we analyze the outputs of SAE, NCAE and DPNE, respectively. We use the randomly sampled subset of 10000 observations from the MNIST data set to verify this purpose. We visualize the 2-dimensional feature space obtained by the SAE, NCAE and the proposed DPNE, as shown in Fig. 1.

As we can see, the proposed DPNE can reorganize the location of each sample in the 2-dimensional plane according to the original data distribution, and these locations successfully reveal the manifold structure of the original data. We also provide the corresponding cluster accuracy. Compare to the SAE and NCAE, it is observed that preserving the distribution of original data can significantly improve the cluster accuracy.

4.2. Comparisons with Other Algorithms

Table 1 shows the clustering results (accuracy and AMI) of different methods. As in the case of DEC and DCN models, the dimension of the low dimensional representation (SAE, NCAE and the proposed DPNE) is set to be 10, i.e., $D = 10$. Compared to k-means, k-means++ can dramatically improve both the speed and accuracy [27]. So we directly employ the k-means++ to cluster the original data without dimensionality reduction.

It is observed from Table 1 that the AMI of our DPNE is consistent on the five data sets and the proposed DPNE can achieve the highest accuracy on four of the five data sets. This suggests the importance of inherent structure in learning the low dimensional representation. We also observe that the NCAE outperforms the SAE, which means the superiority of the meaningful representation idea in extracting the hidden structure. By leveraging the superiority of both the part-based representation and distribution preserving, the proposed DPNE can learn a better discriminative feature.

5. CONCLUSION

In this paper, we present a novel dimensionality reduction method, called distribution preserving network embedding (DPNE). In DPNE, we use the data distribution to approximate the latent geometrical structure embedded in the high dimensional data space. And then the proposed DPNE can learn a meaningful feature which respects to the above distribution. From analysis of the visualization results, the 2-dimensional part-based representation space obtained by the DPHE preserves the structure buried in the original high dimensional data as much as possible. The experimental results on the image and text data sets, also show that the proposed DPNE can learn a more discriminating feature.

References

[1] Richard O Duda, Peter E Hart, and David G Stork, *Pattern classification*, Springer, 2001.

[2] I. T Jolliffe, "Principal component analysis," *Journal of Marketing Research*, vol. 87, no. 100, pp. 513, 2002.

[3] X He, "Locality preserving projections," *Advances in Neural Information Processing Systems*, vol. 16, no. 1, pp. 186–197, 2003.

[4] Geoffrey E. Hinton and Ruslan R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[5] Kamran Ghasedi Dizaji, Amirhossein Herandi, and Heng Huang, "Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization," in *IEEE International Conference on Computer Vision*, 2017, pp. 5747–5756.

[6] Junyuan Xie, Ross Girshick, and Ali Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proceedings of The 33rd International Conference on Machine Learning*, 2016, vol. 48, pp. 478–487.

[7] Bo Yang, Xiao Fu, Nicholas D. Sidiropoulos, and Mingyi Hong, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, vol. 70, pp. 3861–3870.

[8] D Lee, "Learning the parts of objects with nonnegative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[9] E. Hosseini-Asl, J. M. Zurada, and O. Nasraoui, "Deep learning of part-based representation of data using sparse autoencoders with nonnegativity constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 12, pp. 2486–2498, 2016.

[10] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans Pattern Anal Mach Intell*, vol. 33, no. 8, pp. 1548–1560, 2011.

[11] Hong Huang, *Subspaces Versus Submanifolds A Comparative Study of Face Recognition*, Springer Berlin Heidelberg, 2011.

[12] Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *the Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[13] J. Tian, T. Zhang, A. Qin, Z. Shang, and Y. Y. Tang, "Learning the distribution preserving semantic subspace for clustering," *IEEE Trans Image Process*, vol. 26, no. 12, pp. 5950–5965, 2017.

[14] A. Qin, Z. Shang, J. Tian, T. Zhang, Y. Y. Tang, and J. Qian, "Edge-smoothing-based distribution preserving hyperspherical embedding for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 7, pp. 2501–2512, 2018.

[15] Geoffrey E. Hinton and Richard S. Zemel, "Autoencoders, minimum description length and helmholtz free energy," in *International Conference on Neural Information Processing Systems*, 1993, pp. 3–10.

[16] Q. V Le, "Building high-level features using large scale unsupervised learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 8595–8598.

[17] Andrew Ng, "Sparse autoencoder," `https://web.stanford.edu/class/cs294a/sparseAutoencoder_2011new.pdf`, Stanford University, 2011, in CS294A Lecture notes.

[18] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[19] Honglak Lee, Chaitanya Ekanadham, and Andrew Y. Ng, "Sparse deep belief net model for visual area v2," in *International Conference on Neural Information Processing Systems*, 2007, pp. 873–880.

[20] Tu Dinh Nguyen, Truyen Tran, Dinh Q. Phung, and Svetha Venkatesh, "Learning parts-based representations with nonnegative restricted boltzmann machine," in *ACML*, 2013.

[21] Andre Lemme, René Felix Reinhart, and Jochen Jakob Steil, "Online learning and generalization of parts-based image representations by non-negative sparse autoencoders," *Neural Networks*, vol. 33, no. 9, pp. 194–203, 2012.

[22] J Moser, "On the volume elements on a manifold," *Transactions of the American Mathematical Society*, vol. 120, no. 2, pp. 286–294, 1965.

[23] Yizong Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[24] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002, pp. 603–619.

[25] M. Rosenblatt and Y. P. Mack, "Multivariate k-nearest neighbor density estimates," *Journal of Multivariate Analysis*, vol. 9, pp. 1–15, 1979.

[26] Jan Orava, "K-nearest neighbour kernel density estimation, the choice of optimal k," *Tatra Mountains Mathematical Publications*, vol. 50, no. 1, pp. 39–50, 2011.

[27] David Arthur and Sergei Vassilvitskii, "k-means++:the advantages of careful seeding," in *Eighteenth Acm-Siam Symposium on Discrete Algorithms, New Orleans, Louisiana*, 2007, pp. 1027–1035.