The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Deep Embedding for Determining the Number of Clusters

Yiqi Wang,*[1] Zhan Shi,*[2] Xifeng Guo,[1] Xinwang Liu,[1] En Zhu,[1] Jianping Yin[3]
[1]School of Computer, National University of Defense Technology
Changsha, China, 410073, +8618570609961,wangyiqi12a@nudt.edu.cn
[2]Department of Computer Science, University of Texas at Austin, zshi17@cs.utexas.edu
[3]Dongguan University of Technology, jpyin@dgut.edu.cn

Abstract

Determining the number of clusters is important but challenging, especially for data of high dimension. In this paper, we propose Deep Embedding Determination (DED), a method that can solve jointly for the unknown number of clusters and feature extraction. DED first combines the virtues of the convolutional autoencoder and the t-SNE technique to extract low dimensional embedded features. Then it determines the number of clusters using an improved density-based clustering algorithm. Our experimental evaluation on image datasets shows significant improvement over state-of-the-art methods and robustness with respect to hyperparameter settings.

Introduction

The number of clusters K is significant to clustering problems. Most clustering algorithms require K as input to build up precise models based on datasets and to achieve high clustering accuracy. For deep clustering methods, K is also required to train specific deep learning models and to extract embedded features (Xie, Girshick, and Farhadi 2016). However, obtaining a good estimation of K is not trivial. Prior work in determining the number of clusters suffers from the "curse of dimensionality".

In this paper, we present an effective algorithm for predicting the number of clusters in the high-dimensional dataset, called Deep Embedding Determination (DED). DED exploits the advantages of both feature representation learning techniques and density-based clustering algorithms. It learns appropriate embedded features using a deep convolutional autoencoder (CAE) (Masci et al. 2011) and t-SNE (Maaten and Hinton 2008) visualization techniques, and then predicts the number of clusters by using a new density-based clustering algorithm (Rodriguez and Laio 2014).

The contributions of our work are summarized as below: 1) DED is the first work to determine the number of clusters in high dimensional datasets, which goes a step further in the joint problem of dealing with the number of clusters and dealing with the "curse of dimensionality"; 2) DED outputs the number of clusters K of high dimensional datasets, which enables other deep clustering methods to train specific models and to improve the clustering results of existing clustering algorithms; 3) DED provides a better embedded feature space for cluster visualization.

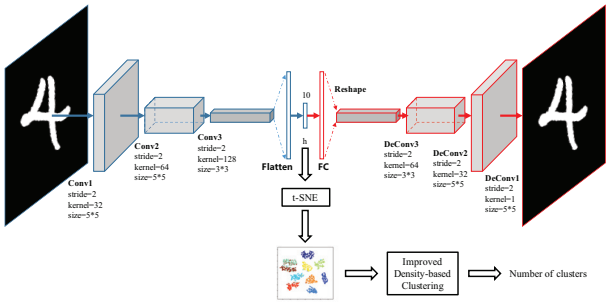

Figure 1: Network Structure of DED

Deep embedding determination

Consider the problem of predicting the number of clusters of a set of points $\{x_i\}_{i=1}^n \subset X$ when the number of clusters K is not known *a priori*. Instead of predicting directly on the data space X, we propose a nonlinear mapping F_w to transform x_i to z_i , where Z is the latent feature space. To make Z appropriate for clustering, the dimension of the feature space Z should be as low as possible. We aim to find a good F_w to produce embedded points with accurate pairwise distance in low dimension and an appropriate clustering algorithm to determine the number of clusters K.

DED has two essential components: the feature extraction model and the number of clusters estimation algorithm. The feature extraction model consists of a CAE and t-SNE. The CAE learns feature representations that preserve intrinsic local structure, and t-SNE further reduces the dimensions of the learned feature while maintaining the pairwise distances. DED uses an improved density-based clustering algorithm to estimate the number of clusters on DED feature space. The network structure of DED is demonstrated in Figure 1.

We propose to transform data to be suitable for estimating the number of clusters in an unsupervised manner in two steps. In the first step, a CAE is trained to extract 10-dimensional features with a reasonable reconstruction loss. Intuitively, the capability of feature representation decreases as the dimension of latent feature decreases. Thus, to obtain

*these authors contribute equally to this work.

Table 1: Comparison of average prediction of MNIST on input data, 10-dimensional CAE features, 2-dimensional t-SNE features , 2-dimensional DED features and 2-dimensional, 10-dimensional features and 30-dimensional features extracted by PCA from input data. "hit rate" represents the ratio of number of correct prediction and the number of all runs (i,e.,10). Note that a range of possible K value has to be provided to the silhouette and gap algorithms, and here we set is as [2,20].

Feature Space	Gap		Silhoutte		DBSCAN		Mean shift		Improved density-based clustering	
	mean	hit rate	mean	hit rate	mean	hit rate	mean	hit rate	mean	hit rate
Input data	19.0 ($\pm$0)	0/10	2.0 ($\pm$0)	0/10	0 ($\pm$0)	0/10	1.0 ($\pm$0)	0/10	1.3k ($\pm$58.5)	0/10
CAE feature	13.9 ($\pm$1.5)	0/10	9.5 ($\pm$1.3)	0/10	0 ($\pm$0)	2/10	1.0 ($\pm$0)	0/10	0.3k ($\pm$36.6)	0/10
T-SNE feature	2.7 ($\pm$0.5)	0/10	11.6 ($\pm$4.0)	0/10	0.8 ($\pm$1.4)	0/10	2.5 ($\pm$0.9)	0/10	8.2 ($\pm$1.2)	3/10
PCA-2 feature	3.0 ($\pm$0)	0/10	3.1 ($\pm$0.3)	0/10	6.0 ($\pm$2.1)	0/10	1.4 ($\pm$0.5)	0/10	31.1 ($\pm$21.9)	0/10
PCA-10 feature	15.1 ($\pm$1.8)	0/10	8.7 ($\pm$2.3)	2/10	0 ($\pm$0)	0/10	1.0 ($\pm$0)	0/10	0.4k ($\pm$35.3)	0/10
PCA-30 feature	18.5 ($\pm$0.9)	0/10	17.0 ($\pm$5.3)	0/10	0 ($\pm$0)	0/10	1.0 ($\pm$0)	0/10	0.6k ($\pm$147.2)	0/10
DED feature	2.9 ($\pm$0.3)	0/10	11.2 ($\pm$1.4)	4/10	4.5 ($\pm$2.4)	1/10	3.6 ($\pm$0.9)	0/10	10.1 ($\pm$0.3)	9/10

appropriate feature representations with reasonable reconstruction loss, the dimension of latent feature space is set to 10 after sufficient experiments. The autoencoder preserves intrinsic local structure of data in 10-dimensional feature space M. However, the features need to be further reduced to lower dimension to fit the improved density-based clustering algorithm. In the second step, DED uses t-SNE algorithm which defines a non-linear mapping from feature space M to a 2-dimensional feature space Z. It minimizes the mismatch between M and Z in terms of pairwise distances by minimizing the non-symmetric difference between the corresponding probability distributions of M and Z. Thus the feature space Z obtained by these two successive steps is more suitable for estimating the number of clusters.

We propose to use the density-based clustering algorithm by (Rodriguez and Laio 2014) with modifications on the density estimation method and an adaptive thresholding mechanism. There are two leading criteria in this method: local density (ρ) and minimum distance with higher density (δ). In replacement of the indicator function, we use the Gaussian kernel in density estimation. Our goal is to find points that have both high ρ and δ, which are taken as cluster centers. We observe that for cluster centers, the ρ is normally much higher than its threshold while the δ can be very close to the threshold in some cases. Instead of using a fixed threshold for δ, we implement a simple and efficient adaptive thresholding method. We consider the points whose ρ are higher than the density threshold as two classes. One class includes the possible cluster centers, and the other class includes the rest data points. We define the separation between these two classes as the ratio of the variance between the classes to the variance within the classes. Then we iteratively change the value of threshold for δ and obtain the best threshold value that achieves the maximal separation value.

Experiments

We evaluate DED on four datasets and compare its ability in determining the right number of clusters against other clustering algorithms. The result is reported in Table 2. We also report the performance of comparing methods on various feature spaces on the MNIST dataset. As shown in Table 1, DED features generally perform better than other features, and the improved density based clustering algorithm can make the best use of DED feature among all methods.

Table 2: Performance of DED in terms of the average number of clusters, variance and the hit rate of the correct predictions among 100 runs.

Dataset	MNIST	USPS	HASY-8	HASY-12
Ground truth	10	10	8	12
Mean	10.1	10.0	8.2	11.7
hit rate	72/100	42/100	49/100	39/100
Variance	0.43	0.86	0.68	1.00

Conclusion and future work

This paper has introduced the Deep Embedding Determination (DED) algorithm, which is able to determine the number of clusters in large datasets of high dimension and outperforms other algorithms by a great margin. However, there exist different criteria for clustering. Learning embedded features for a specific clustering criterion in an unsupervised manner and determining the number of clusters accordingly are challenging and still remain to be explored.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (Project no. 61232016 and 61672528).

References

Maaten, L. v. d., and Hinton, G. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research* 9(Nov):2579–2605.

Masci, J.; Meier, U.; Cireşan, D.; and Schmidhuber, J. 2011. Stacked convolutional auto-encoders for hierarchical feature extraction. *Artificial Neural Networks and Machine Learning–ICANN 2011* 52–59.

Rodriguez, A., and Laio, A. 2014. Clustering by fast search and find of density peaks. *Science* 344(6191):1492–1496.

Xie, J.; Girshick, R.; and Farhadi, A. 2016. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, 478–487.