State of the Art on: Deep Clustering

Francesco Fulco Gonzales, francesco.gonzales@mail.polimi.it

1. Introduction to the research topic

Clustering is a fundamental machine learning problem, whose performance is highly dependent on the quality of data representation. Hence, feature transformations have been extensively used to learn a better data representation for clustering. Deep neural networks are particularly apt to learn non-linear mappings [3] that allow transforming data into a representation that eases the clustering task, without the need of performing manual feature selection and engineering.

Table 1 shows the most prestigious journals and conferences on the research area ranked by relevance metrics such as Impact Score and H5-index.

1.1. Preliminaries

To understand the methods employed to solve deep clustering it is useful to introduce the fundamental neural network architectures used for feature representation.

Multi-Layer Perceptron(MLP) First and simplest type of neural network architecture, it consists of several layers of neurons, such that the output of every hidden layer is the input to next one.

Convolutional Neural Network (CNN) Most useful to analyze visual imagery, if locality and shift-invariance of feature extraction is desired. They are a special case of MLP that restrict connectivity among neurons by taking advantage of the locality of image features. CNNs manage to extract increasingly higher level features from images in the deeper layers, and are therefore very well suited for representation learning on image data.

Deep Belief Network (DBN) Generative graphical models which learn to extract a deep hierarchical representation of the input data. It is composed of several shallow networks such as restricted Boltzmann machines, such that the hidden layer of each sub-network serves as the visible layer of the next sub-network [10].

Journal Name	**Impact Score**
IEEE Transactions on Pattern Analysis and Machine Intelligence	25.25
Pattern Recognition	17.32
IEEE Transactions on Neural Networks and Learning Systems	16.17
Neurocomputing	15.18
Conference Name	**H5-index**
CVPR : IEEE/CVF Conference on Computer Vision and Pattern Recognition	299
ICLR : International Conference on Learning Representations	203
NIPS : Neural Information Processing Systems	198
ICML : International Conference on Machine Learning	171

Table 1: Most prestigious journals and conferences on Machine Learning (source: `www.research.com`). The Impact Score represents the yearly mean number of citations of articles published in the last two years, and the H-index is defined as the maximum value of h such that the journal has published h papers each cited at least h times.

Autoencoder (AE) An autoencoder is a popular architecture that obtains a feasible feature space. It learns a non-linear mapping function by stacking an encoder and a decoder. The encoder network compresses the input to a lower dimension, and the decoder tries to reconstruct the original data from the compressed representation generated by the encoder. The network tries to learn the identity function by minimizing the reconstruction loss, i.e. the error between the reconstructed input and the original one, and doing so it manages to learn a lower dimensional embedding of the input data [16].

Generative Adversarial Network (GAN) [4] A system of two adversarial neural network models that engage in a zero-sum game by competing with each other. The generator generates data from stochastic noise, and the discriminator tries to tell whether it is real (coming from a training set) or fabricated (from the generator network). The overall objective of the network is to minimize the absolute difference of rewards from both networks so that both networks learn simultaneously as they try to outperform each other.

Variational Autoencoder (VAE) VAE [9] can be seen as a generative variant of AE, that combines variational bayesian methods with the flexibility and scalability of neural networks. The most significant difference between a standard autoencoder and a VAE is that the latter imposes a parametrized probabilistic prior distribution over the latent representation, from which, after the network has been trained, new encodings are sampled and decoded to generate new samples.

From the technological point of view there are a few predominant libraries and framework that simplify the development of neural networks and clustering algorithms by hiding most of the implementation details and provide hardware acceleration capabilities. The most popular deep learning Python libraries, by stars on Github are Tensorflow, Keras, PyTorch and Caffe. For a thorough survey of Machine Learning and Deep Learning frameworks refer to [13].

1.2. Research topic

The goal of clustering is to categorize similar data into one cluster based on some similarity measures, however the performance of classical clustering methods is highly dependent on the input data. Different datasets usually require different similarity measures and separation techniques, and high dimensional input data, e.g. images, is not easily clustered. As a result, dimensionality reduction and representation learning have been extensively used alongside clustering in order to map the input data into a feature space where separation is easier. Deep Clustering leverages the intrinsic capability of neural networks to learn better compressed data representations, without relying on human engineered features. Even though it started mostly within the realm of supervised learning, deep learning's success has been successful in the field of unsupervised learning as well. Most deep learning based clustering approaches result in both deep representations and possibly clustering outputs. Deep clustering is nowadays the state-of-the-art approach to solve clustering, which is becoming an even more relevant problem as the size of the datasets increases, and the cost of manually labeling data is still very high.

2. Main related works

2.1. Classification of the main related works

Classical clustering methods are categorized according to the type of clustering loss they minimize, and usually belong to the families of partition-based methods, density-based methods or hierarchical methods (for an exhaustive list refer to [18]). However, since the essence of deep clustering is learning a clustering-oriented representation, it is not appropriate to classify methods according to the clustering loss, which only covers one minor aspect of a more complex method, instead, we should focus on the network architecture used for representation learning and clustering.

It should be noted that Deep Clustering methods all follow the same basic approach: representation learning using DNNs and feeding these representation as input to a specific clustering method. However, as noted by [1], both of these components are made up by different building blocks that can be mixed and matched:

1. Main neural network
   (a) Architecture of main neural network
   (b) Set of deep features used for clustering
2. Neural network losses
   (a) Non-clustering loss
   (b) Clustering loss
   (c) Method to combine the two losses
3. Cluster updates
4. (Optional) Re-run clustering after network training

Some related works classify models by the type of each of the building blocks that make up the model. However, for the sake of clearness and conciseness, it is best to use the network architecture as a grouping dimension, given that it is the most characterizing feature of a Deep Clustering method.

We therefore divide deep clustering algorithms into four categories: AE-based, CDNN-based, VAE-based, and GAN-based deep clustering. Characteristics of each category are discussed and for each it is provided a representative algorithm.

2.2. Brief description of the main related works

In most approaches the main neural network is used to transform the inputs into a latent representation that is subsequently used for clustering. The following neural network architectures have mostly been used for this purpose:

Autoencoder The autoencoder [16] is an unsupervised network architecture that consists of two parts: an encoder function $h = f_\phi(x)$ which maps original data x into a latent representation h, and a decoder that produces a reconstruction $r = g_\theta(h)$, where ϕ and θ denote the parameters of encoder and decoder respectively. The reconstructed representation r is required to be as similar to x as possible, which is achieved by minimizing the reconstruction loss:

$$L_{rec}(\theta, \phi) = \frac{1}{n} \sum_{i=1}^{n} ||x_i - g_\theta(f_\phi(x_i))||^2$$

Deep Clustering Network DCN [19] is a prominent method that employs an autoencoder to learn a feature representation. This method adopts an approach common to several deep clustering methods: instead of performing dimensionality reduction and clustering separately, it optimizes the two tasks jointly, substantially improving performance, as also shown by [15]. Initially DCN pre-trains an autoencoder and then it jointly optimizes the reconstruction loss and the k-means loss.

Clustering DNN CDNNs are characterized by their absence of network loss, since they are only trained using a special clustering loss. The network architecture can be very deep and it can be pre-trained on large-scale image datasets, which have been extensively shown to boost the clustering performance. This finding lead to a reformulation of clustering as a Multi-View Clustering problem where the features extracted from several different pre-trained CNN architectures are seen as views of the same object, which are then concatenated and clustered using other deep clustering methods [6, 5, 7]. Other methods instead are initialized by training a RBM or an autoencoder, and then fine-tune the network with the clustering loss, and yet other methods perform no pre-training at all and still achieve good performance thanks to a well-designed clustering loss.

Deep Embedded Clustering DEC [17] is one of the most well-known methods of deep clustering and is often used as a baseline for publications. It clusters data by simultaneously learning a set of k cluster centers $\{\mu_j \in Z\}_{j=1}^{k}$ in the feature space Z and the parameters θ of the DNN that maps data points into Z. DEC is initialized by the autoencoder, and then it jointly optimizes the network parameters and cluster assignment iteratively until convergence. It performs soft assignment, which means that every cluster assignment comes with its probability of belonging to that cluster, and then uses the highest confidence assignments to learn the clusters that minimize the Kullback–Leibler divergence between their distribution, called auxiliary target distribution, and the whole data distribution. A variant of DEC called Discriminatively Boosted Clustering instead uses fully convolutional autoencoders for image feature learning to improve the performance on image data [11].

Joint Unsupervised Learning JULE [20] also relies on the intuition of using clustering as a supervisory signal for image representation. This method uses a CNN for representation learning and agglomerative clustering for clustering, and optimizes both in a recurrent process that minimizes a unified triplet loss and is trained end-to-end. In the forward pass hierarchical image clustering is performed by merging similar clusters, while in the backward pass feature representation parameters are updated by minimizing the loss generated in the forward pass.

Generative Adversarial Network GAN [4] is a popular deep generative model that can not only perform clustering tasks, but also generate new samples from the obtained clusters. It works by setting up a min-max adversarial game between two networks: a generative network, and a discriminative network. The generative network tries to map a sample z from a prior distribution $p(z)$ to the data space, while the discriminative network tries to compute the probability that an input is a real sample from the data distribution, rather than a sample generated by the generative network. The purpose of doing so is that the generator learns the distribution of data with an arbitrary prior distribution. It should be noted that the main disadvantages of GANs are the difficulty to converge, mode collapse and vanishing gradient [2].

ClusterGAN ClusterGAN [12] proposes a GAN training methodology that clusters in the latent space, and obtains state-of-the-art performance on clustering, as well as good interpretability and interpolation ability. This method overcomes the problem of the poor inherent capability of GANs to cluster, caused by a smooth scattering of its learned representation in the latent space. The solutions consists in (1) using a mix of discrete and continuous latent variables in order to create a non-smooth geometry in the latent space, (2) adopting a novel backpropogation algorithm accommodating the discrete-continuous mixture, as well as an explicit inverse-mapping network to obtain the latent variables given the data points, since the problem is non-convex, and (3) jointly training the GAN along with the inverse-mapping network with a clustering-specific loss so that the distance geometry in the projected space reflects the distance-geometry of the variables.

Variational Autoencoder VAE [9] can be seen as a generative variant of AE, that combines variational bayesian methods with the flexibility and scalability of neural networks. While AE are able to learn the input data distribution and the latent representation, fall short when it comes to learning the latent representation distribution. Therefore to generate data using AE the encodings are sampled randomly and are passed to the decoder to generate a new reconstructed sample, which is unsatisfying, since the chosen embedding was not realistic, i.e. sampled by the original latent distribution. To address this issue, VAE set out to learn the distribution of the encodings by assuming that the encodings belong to a Gaussian distribution of mean μ and variance σ^2. This new parametrization (often called the "reparametrization trick") allows us to sample noisy data from the code distribution.

Variational Deep Embedding VaDE [8] combines VAE [9] and a Gaussian Mixture Model (GMM) [14] for clustering tasks. VaDE models the data generative process by a GMM and a DNN f : 1) a cluster is picked up by the GMM; 2) from which a latent representation z is sampled; 3) DNN f decodes z to an observation x. Moreover, VaDE is optimized by using another DNN g to encode observed data x into latent embedding z, so that the Stochastic Gradient Variational Bayes (SGVB) estimator and the reparameterization trick [9] can be used to maximize the evidence lower bound (ELBO). VaDE generalizes VAE in that a Mixture-of-Gaussians prior replaces the single Gaussian prior. Hence, VaDE is by design more suitable for clustering tasks.

2.3. Discussion

Deep Clustering algorithms are numerous and make use of a wide variety of neural network architectures and losses, each with its own advantages and disadvantages, which are outlined in this section for the four categories discussed above.

Almost all methods apply a clustering loss, specific to the clustering algorithm utilized, since it has been shown to improve performance compared to omitting the clustering loss [17, 19]. Conversely, the absence of a network loss characterizes CDNN-based methods, that optimize the network using only the clustering loss, therefore allowing for deeper networks and supervisedly pre-trained architectures which enables clustering on large-scale image datasets. However, they are also prone to learn a corrupted feature space if the clustering loss is not properly designed. AE-based algorithms' reconstruction loss instead ensures the network learns a feasible representation and avoids obtaining trivial solutions, with the caveat that it limits the network depth due to the high computational load, and that the two losses should be carefully balanced through a hyper-parameter. The peculiarity of VAE- and GAN-based clustering algorithms is the capability of generating samples from the obtained clusters. VAE-based methods have a good theoretical guarantee, as they minimize the variational lower bound on the marginal likelihood of data, but suffer from high computational complexity. GAN-based methods are more flexible and diverse than VAE-based, in fact some of them aim at learning general interpretable representations and treat clustering as a specific application of the framework. Their downside is that they suffer from problems common to GANs, i.e. mode collapse and failure to converge. It is worth noting that VAE- and GAN-based algorithms have a higher computational complexity than AE-based and CDNN-based algorithms, whose complexity is determined by the network architecture (e.g. a deep CNN is more computationally expensive than a simple MLP) and the clustering algorithms (e.g. the cost of agglomerative clustering much higher than the cost of k-means).

References

[1] Aljalbout, E., Golkov, V., Siddiqui, Y., Strobel, M., and Cremers, D. Clustering with deep learning: Taxonomy and new methods. *arXiv preprint arXiv:1801.07648* (2018).

[2] Arjovsky, M., and Bottou, L. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862* (2017).

[3] Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence 35*, 8 (2013), 1798–1828.

[4] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems 27* (2014).

[5] Guérin, J., and Boots, B. Improving image clustering with multiple pretrained cnn feature extractors. *arXiv preprint arXiv:1807.07760* (2018).

[6] Guérin, J., Gibaru, O., Thiery, S., and Nyiri, E. Cnn features are also great at unsupervised classification. *arXiv preprint arXiv:1707.01700* (2017).

[7] Guérin, J., Thiery, S., Nyiri, E., Gibaru, O., and Boots, B. Combining pretrained cnn feature extractors to enhance clustering of complex natural images. *Neurocomputing 423* (2021), 551–571.

[8] Jiang, Z., Zheng, Y., Tan, H., Tang, B., and Zhou, H. Variational deep embedding: An unsupervised and generative approach to clustering, 2017.

[9] Kingma, D. P., and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[10] Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning* (2009), pp. 609–616.

[11] Li, F., Qiao, H., and Zhang, B. Discriminatively boosted image clustering with fully convolutional auto-encoders. *Pattern Recognition 83* (2018), 161–173.

[12] Mukherjee, S., Asnani, H., Lin, E., and Kannan, S. Clustergan: Latent space clustering in generative adversarial networks. In *Proceedings of the AAAI conference on artificial intelligence* (2019), vol. 33, pp. 4610–4617.

[13] Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., Lopez Garcia, A., Heredia, I., Malík, P., and Hluchỳ, L. Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. *Artificial Intelligence Review 52*, 1 (2019), 77–124.

[14] Reynolds, D. A. Gaussian mixture models. *Encyclopedia of biometrics 741*, 659-663 (2009).

[15] Song, C., Liu, F., Huang, Y., Wang, L., and Tan, T. Auto-encoder based data clustering. In *Iberoamerican congress on pattern recognition* (2013), Springer, pp. 117–124.

[16] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res. 11* (dec 2010), 3371–3408.

[17] Xie, J., Girshick, R., and Farhadi, A. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning* (2016), PMLR, pp. 478–487.

[18] Xu, D., and Tian, Y. A comprehensive survey of clustering algorithms. *Annals of Data Science 2*, 2 (2015), 165–193.

[19] Yang, B., Fu, X., Sidiropoulos, N. D., and Hong, M. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning* (2017), PMLR, pp. 3861–3870.

[20] Yang, J., Parikh, D., and Batra, D. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 5147–5156.