주의 메커니즘이있는 자동 인코더를 사용한 딥 연속 텍스트 클러스터링

도트엉동, 백으뜸, 김경백,양형정*
광주광역시 전남대학교 전자컴퓨터공학부
*교신 저자 : hjyang@jnu.ac.kr

Deep Continuous Text Clustering using Autoencoder with Attention Mechanism

Truong-Dong Do, Eu-Tteum Baek, KyungBaek Kim, Hyung-Jeong Yang*
School of Electronics & Computer Engineering, Chonnam National University,
Gwangju, South Korea

요 약

Clustering is one of the fundamental experimental procedures in the analysis of scientific data. Despite more than half a century of research, existing clustering algorithms have limited effectiveness because of the less informative in high-dimensional spaces of interpoint distances. Embedding the data into a lower-dimensional may corrupt features space, which leads to non-representative meaningless features and this hurts clustering performance. In this paper, the attention deep continuous clustering (ADCC) is proposed to preserve data structures and significant features. The deep autoencoder with attention mechanism used to embed the data into a lower-dimensional space and optimized as a part of the clustering process. By integrating the continuous clustering and autoencoder's reconstruction loss, ADCC can jointly optimize cluster labels assignment and learn features at the same time. The proposed model is optimized via stochastic gradient descent and backpropagation. Also, the proposed approach does not rely on prior knowledge of the number of ground truth clusters. Therefore, we avoid discrete reconfigurations of the objective that characterize prior clustering algorithms. The experimental results on texting document datasets demonstrated the importance of attention autoencoder in data structure preservation and the effectiveness of our algorithm.

Keywords: Deep Text Clustering, Continuous Clustering, High Dimensional Clustering, Attention Autoencoder, Jointly Training

1. Introduction

Clustering is a vital research topic in data analysis and machine learning. Well-known approaches include center-based methods and their generalizations [1], [2], and spectral methods [3], [4] group data on handcrafted features according to intrinsic characteristics or similarity. However, when the dimension of input data space is very high, the clustering becomes ineffective due to unreliable similarity metrics [5], [6], [7].

Transforming data from high dimensional feature space to lower-dimensional space in which to perform clustering is an intuitive solution and remains an open problem. This can be done by applying dimension reduction techniques like Principal Component Analysis (PCA), but the representation ability of these shallow models is limited. Thanks to the development of deep learning, such feature transformation can be achieved by using Deep Neural Networks (DNN). We consider this kind of clustering as deep clustering.

Deep clustering is most recently proposed and leaves a lot of unsolved problems. The primitive work in deep clustering focuses on learning features that preserve some properties of data by adding prior knowledge to the subjective [8], [9]. There works combined two-stage: feature transformation and then clustering. Eventually, the goal is to perform nonlinear embedding and clustering jointly. Later, algorithms that jointly accomplish feature transformation and clustering come into being [10], [11]. The Deep Embedded Clustering (DEC) [11] algorithm defines an effective objective in a self-learning manner. This clustering loss is used to update parameters of transforming network and cluster centers simultaneously. The cluster assignment is implicitly integrated into soft labels.

However, the data structure preservation cannot be guaranteed. Thus, the feature transformation may be misguided, leading to the corruption of embedded space. Besides, these algorithms require setting the number of

clusters a priori. And the optimization procedures they employ involve discrete reconfigurations of the objective, such as discrete reassignments of data points to centroids or merging of putative clusters in an agglomerative procedure. Thus, it is challenging to integrate them with an optimization procedure that modifies the embedding of the data itself.

To deal with this problem, inspired by [12], we use autoencoder with attention to learn embedded features and to preserve the local structure of data generating distribution. We propose to incorporate autoencoder into deep continuous clustering (DCC) [13] framework, a recent formulation of clustering as this way, the proposed framework can jointly perform clustering and learn representative features with feature structure preservation. We refer to our algorithm as attention deep continuous clustering (ADCC). The optimization of ADCC can directly perform minibatch stochastic gradient descent and backpropagation. The experimental results validate our assumption and the effectiveness of our ADCC.

2. Proposed Method

Consider a dataset X = [x1, . . ., xN] with N samples is a set of points in R^D that must be clustered. When D is high, most clustering algorithms do not operate effectively. To overcome this problem, we embed the data into a lower-dimensional space R^d. An autoencoder is a neural network that is trained to attempt to copy its input to its output. Internally, it has a hidden layer z that describes an embedded code used to represent the input. The network consists of two parts: an encoder function $z = f_\varphi(x)$ and a decoder $r = g_\theta(z)$ that produces a reconstruction. The reconstructed representation r is required to be as similar to x as possible. When the distance measure of two variables is mean square error, given a set of data samples $\{x_i\}_{i=1}^{N}$, it tries to minimize the following reconstruction loss:

$$\min_{\varphi,\theta} L_{rec} = \min \frac{1}{N}\sum_{i=1}^{N} \left\| x_i - g_\theta\left(f_\varphi(x_i)\right)\right\|^2 \qquad (1)$$

The input with hundreds of dimensional represented by several dimensional space will surely lead to information loss, inadequate translation. To deal with this problem, attention mechanism [14] is used to plug a context vector into the gap between encoder and decoder. By utilizing this, it is possible for the decoder to capture global information rather than solely to infer based on one hidden state. The attention weights, context vector and attention vector are computed as following formulas respectively:

$$\alpha_{ts} = \frac{exp\left(score\left(h_t, \bar{h}_s\right)\right)}{\sum_{s'=1}^{S} exp\left(score\left(h_t, \bar{h}_{s'}\right)\right)} \qquad (2)$$

$$c_t = \sum_s \alpha_{ts}\bar{h}_s \qquad (3)$$

$$a_t = f(c_t, h_t) = tanh\left(W_c[c_t; h_t]\right) \qquad (4)$$

The Attention Deep Continuous Clustering (ADCC) algorithm optimizes the following objective [13]:

$$\mathcal{L}(\Omega, Z) = \underbrace{\frac{1}{D}\|X - G_\omega(Y)\|_F^2}_{\text{reconstruction loss}} + \frac{1}{d}\left(\underbrace{\sum_i \rho_1(\|z_i - y_i\|_2; \mu_1)}_{\text{data loss}} + \underbrace{\lambda\sum_{(i,j)\in\varepsilon} w_{i,j}\,\rho_2(\|z_i - y_i\|_2; \mu_2)}_{\text{pairwise loss}}\right) \qquad (5)$$

where $Y = F_\theta(X)$. In (5), the mapping function F_θ and G_ω are performed by an autoencoder. The graph ε is constructed on X using the mutual kNN criterion[15], augmented by the minimum spanning tree to ensure co nnectivity to all datapoints. The role of M-estimators ρ_1 and ρ_2 is to pull the representatives of a true under -lying cluster into a single point, while disregarding sp urious connections across clusters.

The parameters μ_1 and μ_2 control the radii of the c onvex basins of the estimators. The weights $\omega_{i,j}$ are se t to balance the contribution of each data point to the p airwise loss. The parameter λ set the balance between the data loss and the pairwise loss. To balance the diff erent terms, we set $\lambda = \frac{\|Y\|_2}{\|A\|_2}$, where A = $\sum_{(i,j)\in\varepsilon} w_{i,j}(e_i - e_j)(e_i - e_j)^T$ and $\|\cdot\|_2$ denotes the spectral norm.

Objective (5) can be optimized using scalable moder n forms of stochastic gradient descent (SGD). The zi is updated only via its corresponding loss and pairwise ter ms. On the other hand, the autoencoder parameters Ω are updated via all data samples. Thus, in a single epoc h, there is bound to be a difference between the update rates for Z and Ω. To deal with this imbalance, an ada ptive solver such as Adam [16] is used.

3. Experimental Results

3.1. Datasets.

We conduct experiments on RCV1 datasets [17]. This is a document dataset contains around 810,000 Reuters newswire articles. Following DCC [13], only the four root categories: corporate/industrial, government/social, markets and economics are considered, and all articles with multiple labels are pruned. We report results on a randomly sampled subset of 10,000 articles. TF-IDF features on the 2,000 most frequently occurring word stems are computed and normalized to the range [0, 1]. The sampled dataset is referred as to RCV1-10K.

Note that ADCC is an unsupervised learning algorithm. Unlabeled data is embedded and clustered with no supervision. There is thus no train/test split.

The distribution of RCV1-10K dataset is illustrated in Fig.1, the imbalance, defined as the ratio of the largest and the smallest cardinalities of ground-truth clusters, are shown as 5.79. Fig.2 shows the visualization of the data points of RCV1-10K dataset by t-SNE [18].

3.2. Implementation

We report experimental results for FCN-DCC and ADCC. For fully-connected autoencoders, we use the same autoencoder architecture as DEC [19] with the following dimensions: D-500-500-2000-d-2000-500-500-D.

DCC uses three hyperparameters: the embedding dimensionality d, the nearest neighbor parameter k for mkNN graph construction, and the update period M for graduated nonconvexity. In this experiment, we config d = 10, k = 10 (the setting used in [20]), M = 20 and the cosine distance metric is used for graph construction.

For autoencoder initialization, a minibatch size of 256 and a dropout probability of 0.2 are set. During the optimization using the DCC objective, the Adam solver is used with its default learning rate of 0.001 and momentum 0.99. Minibatches are constructed by sampling 128 edges. ADCC was implemented using the PyTorch library.

3.3. Measures

Common measures of clustering accuracy include normalized mutual information (NMI)[21] and clustering a

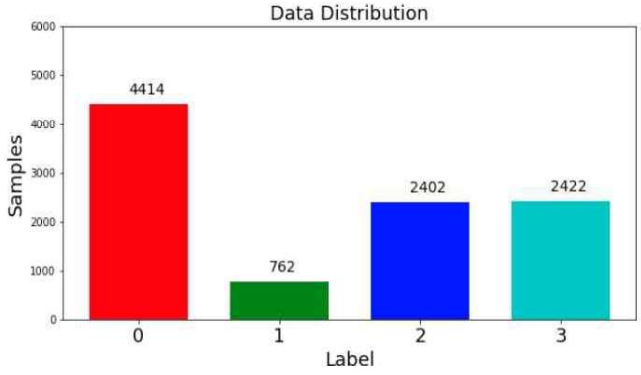

Figure 1. The distribution of the RCV1-10K dataset.

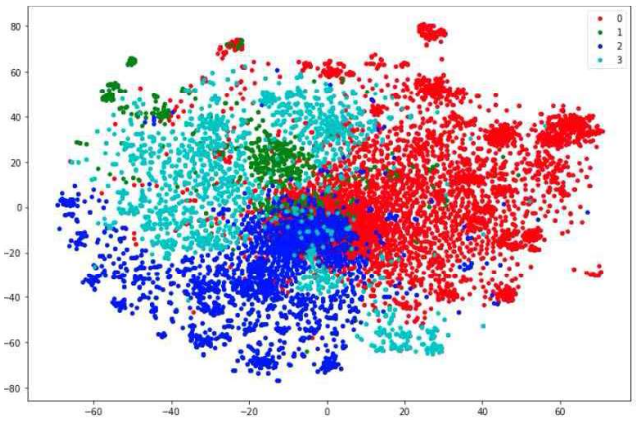

Figure 2. The randomly sampled subset of 10K points from the RCV1 dataset, visualized using the t-SNE.

ccuracy (ACC). However, NMI is known to be biased in favor of fine-grained partitions and ACC is also biased on imbalanced datasets[22]. To overcome these biases, we use adjusted mutual information (AMI), defined as:

$$AMI(c, \hat{c}) = \frac{MI(c, \hat{c}) - \mathrm{E}[MI(c, \hat{c})]}{\sqrt{H(c)H(\hat{c})} - \mathrm{E}[MI(c, \hat{c})]} \quad (6)$$

Here $\mathrm{H}(\cdot)$ is the entropy, $\mathrm{MI}(\cdot,\cdot)$ is the mutual information. c and $\hat{c}$ are the two partitions being compared. AMIlies in a range [0, 1]. Higher is better. For completeness, results according to ACC and NMI are also reported.

3.4. Results

The results with two methods, fully-connected DCC (FCN-DCC) and proposed ADCC are reported in Table. 1. The FCN-DC, which configured with the same parameters get lower performance compared to ADCC.

Table 1. The experimental result on the RCV1-10K dataset

Model	AMI	NMI	ACC
FCN-DCC	0.495	0.498	0.563
ADCC (proposed method)	0.511	0.515	0.592

4. Conclusion

This paper proposed Attention Deep Continuous Clustering (ADCC) algorithm, which jointly performs dimensionality reduction and clustering by optimizing a global continuous objective using scalable gradient-based solvers. Dimensionality reduction remained the data structure by incorporating an autoencoder with the attention mechanism. The embedding was optimized as a part of the clustering process and the resulting network produces clustered data. The presented approach did not rely on a priori knowledge of the number of ground truth clusters.

Empirical experiments demonstrated that structure preservation is vital to a deep clustering algorithm and can favor clustering performance.

Our future work includes extending this method to improve the accuracy of ADCC by improving the preprocessing stage and extract contextual word embedding of texting data before doing the dimensionality reduction.

In this paper, we proposed a confidence of adaptive ranging technique guided by deep transfer learning model for group-based cohesion prediction in the wild. The main contribution of the paper lied in the use of CAR to scale the cohesive score to various range and multi-task learning from a single image. For future work, we plan to explore other cues of the image for a more robust cohesion prediction.

Acknowledgement

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2019-2016-0-00314) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

참 고 문 헌

[1] A. Bagnall and G. Janacek, "Clustering time series with clipped data," Mach. Learn., vol. 58, no. 2 - 3, pp. 151 - 178, 2005.

[2] M. Teboulle, "A unified continuous optimization framework for center-based clustering methods," J. Mach. Learn. Res., vol. 8, pp. 65 - 102, 2007.

[3] U. von Luxburg, "A Tutorial on Spectral Clustering," Nov. 2007.

[4] Y. Ng, Andrew Y., Jordan, Michael I., and Weiss, "On spectral clustering: Analysis and an algorithm.," Neural Inf. Process. Syst., no. January, p. 14, 2001.

[5] K. Ball, "An elementary introduction to modern convex geometry," Flavors Geom., vol. 31, no. 52, pp. 24 - 27, 1997.

[6] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When Is 'Nearest Neighbor' Meaningful?," Springer, Berlin, Heidelberg, 1999, pp. 217 - 235.

[7] M. Steinbach, M. Steinbach, L. Ertöz, and V. Kumar, "The challenges of clustering high-dimensional data," NEW VISTAS Stat. Phys. Appl. ECONOPHYSICS, BIO INFORMATICS, PATTERN Recognit., 2003.

[8] F. Tian, B. Gao, Q. Cui, E. Chen, and T.-Y. Liu, "Learning Deep Representations for Graph Clustering," AAAI, pp. 1293 - 1299, 2014.

[9] X. Peng, S. Xiao, J. Feng, W. Y. Yau, and Z. Yi, "Deep subspace clustering with sparsity prior," IJCAI Int. Jt. Conf. Artif. Intell., vol. 2016-Janua, pp. 1925 - 1931, 2016.

[10] J. Yang, D. Parikh, and D. Batra, "Joint Unsupervised Learning of Deep Representations and Image Clusters," Apr. 2016.

[11] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised Deep Embedding for Clustering Analysis," vol. 48, 2015.

[12] A. Vaswani et al., "Attention Is All You Need," Jun. 2017.

[13] S. A. Shah and V. Koltun, "Deep continuous clustering," Proc. Natl. Acad. Sci., vol. 114, no. 37, pp. 9814 - 9819, 2017.

[14] M. T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," Conf. Proc. - EMNLP 2015 Conf. Empir. Methods Nat. Lang. Process., pp. 1412 - 1421, 2015.

[15] M. R. Brito, E. L. Chávez, A. J. Quiroz, and J. E. Yukich, "Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection," Stat. Probab. Lett., vol. 35, no. 1, pp. 33 - 42, Aug. 1997.

[16] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Dec. 2014.

[17] D. D. Lewis, Y. Yang, T. G. Rose, F. Li, and F. Li LEWIS, "RCV1: A New Benchmark Collection for Text Categorization Research," 2004.

[18] L. Van Der Maaten and G. Hinton, "Visualizing Data using t-SNE," 2008.

[19] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised Deep Embedding for Clustering Analysis," Nov. 2015.

[20] S. A. Shah and V. Koltun, "Robust continuous clustering," Proc. Natl. Acad. Sci., vol. 114, no. 37, pp. 9814 - 9819, Sep. 2017.

[21] A. Strehl and J. Ghosh, "Cluster Ensembles-A Knowledge Reuse Framework for Combining Multiple Partitions," 2002.

[22] N. Xuan Vinh, J. Epps, and J. Bailey, "Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance," 2010.