DEEP ADVERSARIAL GAUSSIAN MIXTURE AUTO-ENCODER FOR CLUSTERING

Warith Harchaoui
Research and Development
Oscaro.com
Paris, France
warith.harchaoui@oscaro.com

Pierre-Alexandre Mattei & Charles Bouveyron
MAP5, UMR 8145
Université Paris-Descartes
Sorbonne Paris Cité
{pierre-alexandre.mattei,charles.bouveyron}
@parisdescartes.fr

ABSTRACT

Feature representation for clustering purposes consists in building an explicit or implicit mapping of the input space onto a feature space that is easier to cluster or classify. This paper relies upon an adversarial auto-encoder as a means of building a code space of low dimensionality suitable for clustering. We impose a tunable Gaussian mixture prior over that space allowing for a simultaneous optimization scheme. We arrive at competitive unsupervised classification results on hand-written digits images (MNIST) that is customarily classified within a supervised framework.

1 INTRODUCTION

The computer vision field has recently witnessed major progress thanks to end-to-end deep-learning systems since the seminal work of (LeCun et al., 1990) and more recently (Krizhevsky et al., 2012). Most of the work however has been carried out in a supervised context. Our effort leverages that wealth of existing research but for clustering in a unsupervised framework with adversarial auto-encoders (Makhzani et al., 2015) inspired by generative adversarial networks (Goodfellow et al., 2014).

2 RELATED WORK

Clustering is one of the most fundamental tasks in machine learning (Duda et al., 2012). This task consists in grouping similar objects together without any supervision, meaning there are no labels to guide the grouping. The goal is to find structure in data to facilitate further analysis. It is a desirable goal in data analysis, visualization and is often a preliminary step in many algorithms for example in Computer Vision (Ponce & Forsyth, 2011).

In this paper, the clustering is achieved in the code space through a combination of auto-encoders (Vincent et al., 2010) and a Gaussian mixture model (McLachlan & Peel, 2004). We propose an algorithm to perform unsupervised clustering with the adversarial auto-encoder framework that we call "DAC" for Deep Adversarial Clustering. Indeed, we postulate that a generative approach, namely a tuned Gaussian mixture model, can capture an explicit latent model that is the origin of the observed data, in the original space.

3 DEEP ADVERSARIAL GAUSSIAN MIXTURE AUTO-ENCODERS FOR CLUSTERING

In our approach, the embedding step role is to make the data representation easier to classify than in the initial space. We have chosen GMM for its nice theoretical properties (Fraley & Raftery, 2002; Biernacki et al., 2000). Moreover, it is well-known that such clustering algorithms work better in low-dimensional settings which motivates our use of dimension reduction performed here by the auto-encoder.

In clustering, we have a dataset of points $(\mathbf{x}_1, ..., \mathbf{x}_i, ..., \mathbf{x}_n)$ where each datapoint $\mathbf{x}_i$ lives in a D-dimensional space. First, we build an auto-encoder that consists of neural-network-based encoder $\mathcal{E}$ and decoder $\mathcal{D}$ parametrized by $\theta_{\mathcal{E}}$ and $\theta_{\mathcal{D}}$ respectively. The encoder $\mathcal{E}$ maps the data points from their original space to a code d-dimensional space ($d < D$). The decoder $\mathcal{D}$ maps them back from the code space to the original one, in such a way that each datapoint $\mathbf{x}_i$ is roughly reconstructed through the encoder and decoder: $\mathcal{D}(\mathcal{E}(\mathbf{x}_i)) \simeq \mathbf{x}_i$. The idea is that if the reconstruction is viable then we have compressed the information of each example without too much loss.

Second, similarly to the work of Makhzani et al. (2015), we add up an adversarial part to the system with: *(i)* a Gaussian-mixture-based random generator $\mathcal{H}$ whose proportions (π_k), means (μ_k) and covariance matrices (Σ_k) for $k = 1, ..., K$ are parametrized by $\theta_{\mathcal{H}}$. An instance of such generated random vectors is noted $\mathbf{z}_i$ and lives in the same code d-dimensional space as above; *(ii)* a neural-network-based adversarial discriminator $\mathcal{A}$ with weights and biases parametrized by $\theta_{\mathcal{A}}$ whose role is to continously force the code space to follow the Gaussian mixture prior.

Finally, we get three objectives optimized through a Stochastic Gradient Descent (SGD) (see (Bottou & LeCun, 2004)) scheme with Back-Propagation (BP) (see (LeCun et al., 2012)):

1. the traditional auto-encoder reconstruction objective to minimize over $\theta_{\mathcal{E}}$ and $\theta_{\mathcal{D}}$:

$$\mathcal{L}_R(\theta_{\mathcal{E}}, \theta_{\mathcal{D}}) = \sum_{i=1}^{n} \|\mathbf{x}_i - \mathcal{D}(\mathcal{E}(\mathbf{x}_i)\|_2^2 \tag{1}$$

One could have chosen a cross-entropy loss function but it does not make any conceptual difference except that it does not work for non-normalized data. In fact, one could use any differentiable distance;

2. the Gaussian mixture model likelihood to maximize over $\theta_{\mathcal{H}}$ for K clusters/modes:

$$\ell_{GMM}(\theta_{\mathcal{H}}) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathcal{E}(\mathbf{x}_i), \mu_k, \Sigma_k) \tag{2}$$

where, $\mathcal{N}$ is the normal Gaussian distribution. But in practice, we will prefer its logarithm for numerical stability reasons (which is common with GMMs):

$$\mathcal{L}_{GMM} = \log(\ell_{GMM}) \tag{3}$$

3. the adversarial objective to optimize a $\min_{\theta_{\mathcal{A}}} \max_{\theta_{\mathcal{E}}}$ problem:

$$\mathcal{L}_A = \frac{-1}{2n} \left(\sum_{i=1}^{n} \log_e(\mathcal{A}(\mathcal{E}(\mathbf{x}_i))) + \sum_{i=1}^{n} \log_e(1 - \mathcal{A}(\mathbf{z}_i)) \right) \tag{4}$$

$\mathcal{L}_A$ is the cross-entropy loss with the true codes $\mathcal{E}(\mathbf{x}_i)$ from data as positive examples and fake generated codes $\mathbf{z}_i$ from the random generator as negative examples.
For Eq. (4), $\mathbf{z}_i$ comes from from the random generator $\mathcal{H}$. Indeed, $\mathbf{z}$ is a random vector from the multimodal Gaussian distribution $\mathcal{H}$ defined above by the πs, μs and Σs:

$$p(\mathbf{z}|\theta_{\mathcal{H}}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{z}, \mu_k, \Sigma_k) \tag{5}$$

Note that it is possible to generate such vectors using a multinomial random number and the Cholesky decomposition of covariance matrices (Bishop, 2006, p. 528).

4 Experiments

For our empirical experiments, we used the MNIST dataset of 70000 digits images in 10 groups. Throughout the experiments[1], we used the same architecture from Xie et al. (2015) and Jiang et al. (2016) D-500-500-2000-d (D is the dimensionality of the input space *e.g.* 784 for MNIST and $d = 10$ is the dimensionality of the code space) for the encoder for fair comparaisons. Furthermore, we use a d-100-10-1 neural network architecture for the discriminator that, thus, takes a code of dimension d as input and one probability as output of coming from a real data point or a generated random vector.

[1] done in Tensorflow (Abadi et al., 2015) and Scikit-learn (Pedregosa et al., 2011) in Python soon available

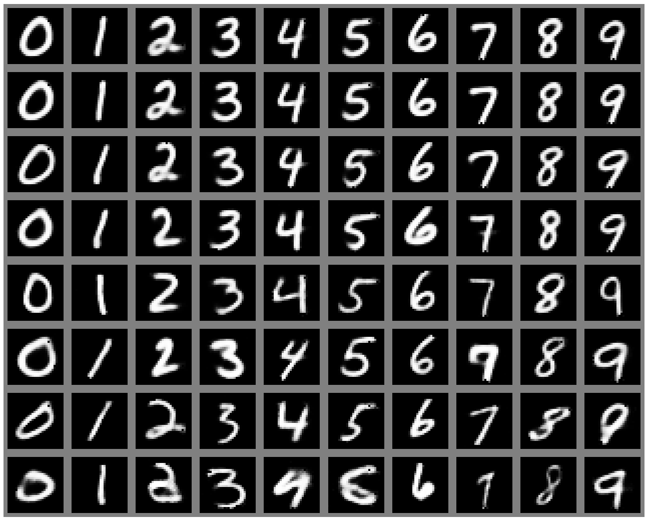

Figure 1: Generated digits images. From left to right, we have the ten classes found by DAC and ordered thanks to the Hungarian Algorithm. From top to bottom, we go further and further in random directions from the centroids (the first row being the decoded centroids).

4.1 Results

The results of DAC compare favorably to the state-of-the-art in Table (1). We also get an extra-boost of accuracy thanks to an Ensemble Clustering method (Weingessel et al., 2003) that combines multiple outputs coming from multiple random initializations.

Datasets	MNIST-70k
DAC EC (Ensemble Clustering over 10 runs)	**96.50**
DAC (median accuracy over 10 runs)	94.08
VaDe (Jiang et al., 2016)	94.06
DEC (Xie et al., 2015)	84.30
GMM	53.73

*: Results taken from (Jiang et al., 2016)

Table 1: Experimental Accuracy Results (%, the higher, the better) based on the Hungarian Method (Kuhn, 1955; Stephens, 2000)

Our experiments show that an auto-encoder dramatically improves all clustering results as (Xie et al., 2015) and (Jiang et al., 2016) are based on auto-encoders. Furthermore, our adversarial contribution outperforms all the previous algorithms.

On the first row of Fig. (1) we show the decoded GMM centroids and each corresponds to a cluster of digits. On the other rows, we go further and further from the centroids and we can see the style of the digits becoming fancier along with the vertical axis.

5 Conclusion

Within the context of the algorithm laid out above, some symbiosis does operate between clustering and non-linear embedding while preserving the reconstruction ability. There are improvements that can be made mainly to overcome problems in the adversarial part, the online Gaussian mixture model and better auto-encoders.

Acknowledgments

The authors would like to thank Oscaro.com for the funding and also Vincent Delaitre and Ross Girshick for fruitful discussions.

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `http://tensorflow.org/`.

Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Pattern Analysis and Machine Intelligence*, 22 (7):719–725, 2000. doi: 10.1109/34.865189.

Christopher M. Bishop. Pattern recognition. *Machine Learning*, 128, 2006.

Léon Bottou and Yann LeCun. Large scale online learning. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf (eds.), *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern classification*. John Wiley & Sons, 2012.

Chris Fraley and Adrian E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, 97(458):611–631, 2002.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: A generative approach to clustering. 2016. URL `http://arxiv.org/abs/1611.05148`.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Harold W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems 2, NIPS 1989*, pp. 396–404. Morgan Kaufmann Publishers, 1990.

Yann A. LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pp. 9–48. Springer, 2012.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Geoffrey McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. URL `http://scikit-learn.org/`.

Jean Ponce and David Forsyth. *Computer vision: a modern approach*. 2011.

Matthew Stephens. Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):795–809, 2000.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

Andreas Weingessel, Evgenia Dimitriadou, and Eurt Hornink. An ensemble method for clustering. In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*, 2003.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. *arXiv preprint arXiv:1511.06335*, 2015.