Final Project VaDE

Name : Van Nguyen , Preethal Reddy Yellareddygari

**I Introduction .**

The goal of this project is to reproduce the result from the paper "Variational Deep Embedding: A Generative Approach to Clustering" . As proposed we also experiement with :

- Convolution architecture
- Using V-metric for clustering evaluation.

**II Methodology**

**A. VaDE :**

The VaDE model is an advancement of the VAE architecture that helps in clustering the latent space representation. In the original VAE model, we obtain the latent space from the encoder, which helps us to create more meaningful data reconstruction from the latent space. However, we still need to make the latent representation more complex. The VaDE model provides us with an approach to clustering the z space by clustering that distribution. This separates the whole z distribution into k clusters in the z distribution.

To implement the VaDE model, we need to initialize the Encoder and Decoder similar to what we do in VAE. We also need to initialize the learning parameter for the GMM model, which is (pi, mu, covariance). The forward pass of VaDE is similar to VAE. We will need to get the z space from the encoder and use it for the decoder. However, the difference in this model is that during training, we try to learn clustering the z space to pre-define k clusters. This helps the distribution to have a clear understanding of the z space at a given input.

To generate a sample $x$ , select a cluster $c$ with corresponding parameters $\mu_c$ and $\sigma_c$. From this cluster, draw $z$ from the Gaussian distribution.

$z \sim \mathcal{N}(\mu_c, \sigma_c^2\mathbf{I})$

If  is binary, let $\mu_x = f(z;\theta)$ and $x$ be sampled from a Bernoulli distribution.

$x \sim \text{Ber}(\mu_x)$

If $x$ is real-valued, set $[\mu_x;\ \ln\sigma_x^2] = f(z;\theta)$ , and sample $x$ from the Gaussian distribution .

$x \sim \mathcal{N}(\mu_x, \sigma_x^2\ \mathbf{I})$

The joint probability model is defined as:

$p(x, z, c) = p(x|z)p(z|c)p(c)$

Where:

$p(c) = \text{Cat}(c|\pi)$

$p(z|c) = \mathcal{N}(z|\mu_c, \sigma_c^2\mathbf{I})$

$p(x|z) = \text{Ber}(x|\mu_x)$ or $\mathcal{N}(x|\mu_x, \sigma_x^2\mathbf{I})$

Here, $\text{Cat}(c|\pi)$represents the categorical distribution parameterized by $\pi$,$\mu_c$ ,$\sigma_c^2$ and are the mean and variance of the Gaussian distribution corresponding to the cluster . $\mathbf{I}$ denotes the identity matrix. The distributions $\text{Ber}(\mu_x)$ , $\mathcal{N}(\mu_x, \sigma_x^2)$represent the multivariate Bernoulli and Gaussian distributions, respectively.

**B ELBO loss function .**

In the VAE model, we need to include two terms in the loss function: the reconstruction loss and the KL loss. The reconstruction loss helps us train the model to recreate results that are close to the original dataset, while the KL loss helps us learn a better representation of the z-space for the decoder. VaDE also follows the same implementation, but it includes an additional term since it has an added component to the model, which is the GMM.
In the Variational Autoencoder with Discrete Latent Variables (VaDE), the log-likelihood of observing data $x$ can be challenging to compute directly.
The initial expression for the log-likelihood involves integrating over all possible latent states:

$$\ln p(x) = \ln \int \sum_c p(x,z,c)\, dz$$

Using Jensen's Inequality, we approximate this log-likelihood by the expected log-ratio of the joint distribution to the posterior distribution of latent variables:

$$\ln p(x) \geq \mathbb{E}_{p(z,c|x)}\left[\ln \frac{p(x,z,c)}{p(z,c|x)}\right]$$

This approximation, known as the Evidence Lower Bound (ELBO), facilitates optimization by providing a surrogate objective that is computationally feasible to maximize.
In the VaDE model, similar to the Variational Autoencoder (VAE), the true posterior $p(z,c|x)$ is intractable. Instead, we use a variational approximation $q(z,c|x) = q(z|x)q(c|x)$ to estimate it.
Expanding the above formula yields:

$$\text{ELBO}(x) = \mathbb{E}_{q(z,c|x)}[\log p(x|z) + \log p(z|c) + \log p(c) - \log q(z|x) - \log q(c|x)]$$

the Evidence Lower Bound (ELBO) can be reformulated using the equations above as:

$$\begin{aligned}\text{ELBO}(x) = \ & \frac{1}{L}\sum_{l=1}^{L}\left(\sum_{i=1}^{D} x_i \ln \mu_{x|i}^{(l)} + (1-x_i)\ln(1-\mu_{x|i}^{(l)})\right) \\ & -\frac{1}{2}\sum_{c=1}^{C}\gamma_c\left(\sum_{j=1}^{J}\ln \sigma_{c|j}^2 + \frac{\tilde{\sigma}_{|j}^2}{\sigma_{c|j}^2} + \frac{(\tilde{\mu}_{|j}-\mu_{c|j})^2}{\sigma_{c|j}^2}\right) \\ & +\sum_{c=1}^{C}\gamma_c \ln\frac{\pi_c}{\gamma_c} + \frac{1}{2}\sum_{j=1}^{J}(1+\ln\tilde{\sigma}_{|j}^2)\end{aligned}$$

In this formulation, $L$ is the training batch size, $D$ represents the dimensionality of input $x$ and output $\mu_x^{(l)}$ , $C$ is the number of clusters, $J$ denotes the dimensionality of latent space, $\pi_c$ is the probability of cluster $c$, $\gamma_c$ represents $q(c|x)$.
This version of ELBO highlights the contribution of each part of the model, from data reconstruction to the regularization by the latent space and cluster distributions.

**III Implementation.**

We use the ELBO loss function as described in section II.B. In order to predict the output cluster, we fit the gmm model with the latent space produced from the desired VaDE model.

- For 1D, we build with a sequence of linear operations.
- For 2D (CNN), we build with a sequence of 2D convolutions with stride = 2 for down-sampling and 2D transpose convolutions with stride = 2 for upsampling.

## IV Evaluation Metric

**V-measure**:  Evaluates the quality of clustering based on the harmony between homogeneity and completeness:

- **Homogeneity** checks if each cluster contains only members from a single class.
- **Completeness** assesses if all members of a given class are in a single cluster.

The V-measure is the harmonic mean of homogeneity and completeness:

$$\text{V-measure} = 2 \times \frac{\text{Homogeneity} \times \text{Completeness}}{\text{Homogeneity} + \text{Completeness}}$$

This metric is useful for evaluating clustering without relying on the actual labels, focusing instead on the intrinsic grouping quality.

## V Result .

| V- measure | Train | Test |
|---|---|---|
| MNIST model 300 epochs | 71.66 | 71.59 |
| MNIST model 25  epochs | 67.1 | 67.5 |
| VaDE CNN 25 epochs C10 | 82.12 | 82.67 |
| VaDE CNN 100 epochs C10 | 84.7 | 85.02 |
| VaDE CNN 25 epoch C14 | 78.63 | 77.89 |

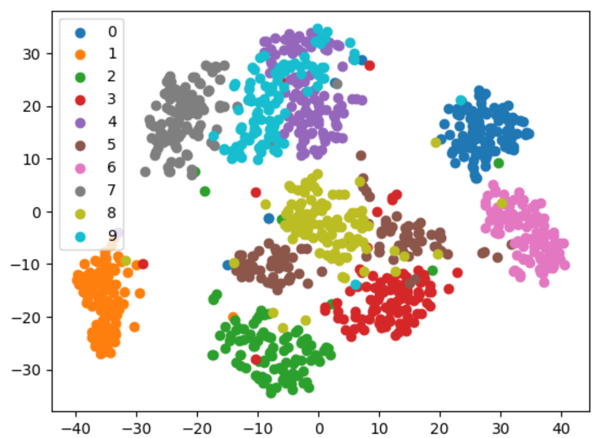


**Figure 1:**  VaDE 25

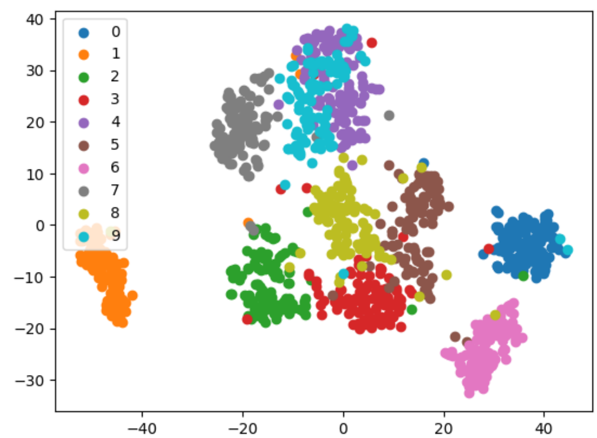


**Figure 2:** VaDE 300

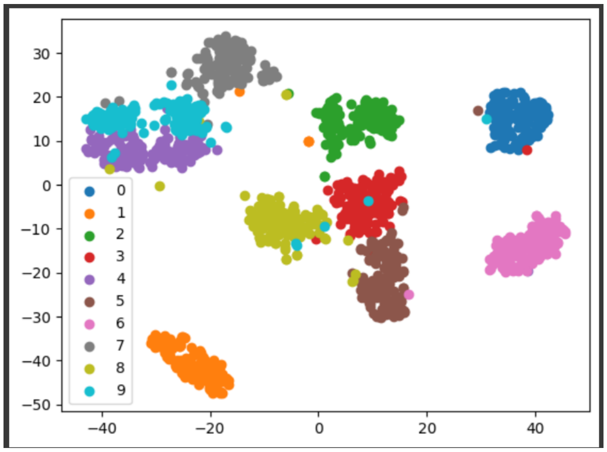


**Figure 3:** VaDE-CNN 25 epochs

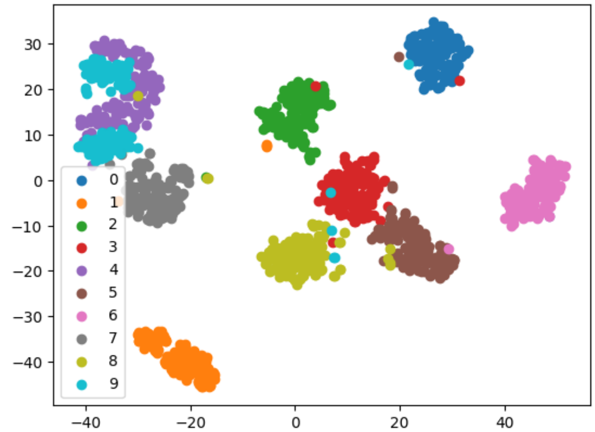


**Figure 4:** VaDE-CNN 100 epochs

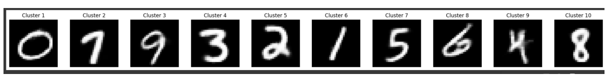

**Figure 5:** CNN 100 epochs 10 cluster

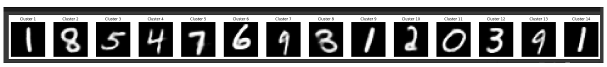

**Figure 6:** 14 cluster CNN 25 epochs

**VI Conclusion**

Variational Deep Embedding (VaDE) effectively combines Variational Autoencoders (VAEs) and Gaussian Mixture Models (GMMs) to enhance unsupervised clustering and data generation. By leveraging the SGVB estimator and reparameterization trick, VaDE optimizes the Evidence Lower Bound, boosting model performance. Implementations using CNNs further improve results, outperforming traditional AE-based models. Additionally, employing V-measure as the clustering metric provides a deeper evaluation of cluster quality. Overall, VaDE demonstrates superior performance in generating realistic samples and precise clustering, establishing itself as a leader in unsupervised learning.

**VII Challenges**

During the project , our primary focus was on understanding the mathematics of ELBO and to implements the math appropriately so that the training can learning to explain the latent space representative with clustering . Beside that , The computation demands of this process were significant, with each experiments( 1D and 2D operations implementation )  we attempt cost over 100 compute units on Google Collabs ranging from debugging to final results .

**VIII Future Work**

In future studies, we plan to explore the performance of the VaDE model across a broader range of datasets. This will help us assess its adaptability and effectiveness in various data environments and further refine its clustering capabilities.

Refferences


- M. (n.d.). VaDE/main.py at master · mori97/VaDE. GitHub. https://github.com/mori97/VaDE/blob/master/main.py
- S. (n.d.). GitHub - slim1017/VaDE: Python code for paper - Variational Deep Embedding：A Generative Approach to Clustering. GitHub. https://github.com/slim1017/VaDE
- jiang, Zhuxi & Zheng, Yin & Tan, Huachun & Tang, Bangsheng & Zhou, Hanning. (2016). Variational Deep Embedding: A Generative Approach to Clustering.
- [1]W. Zhu, “Improving the Applicability of Variational Deep Embedding in Unsupervised Large-Scale Data Clustering,” Dissertation, 2020.