From ”big data” to ”small data” A methodology for unsupervised clustering

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this work, motivated by the ever-increasing demands for clustering accuracy, cybersecurity, limited communication bandwidth and low-power consumption, we propose a new methodology for performing unsupervised clustering on compressed data domains. The main idea is to compress high-dimensional massive data into small-sized data and perform inference based on the compressed data. Once jointly trained for compression and post-compression clustering, the model can be decomposed into two parts: a data vendor that encodes the raw data into compressed data, and a data consumer that classifies the received (compressed) data into classes. In this way, the data vendor benefits from data security and communication bandwidth, while the data consumer benefits from low computational complexity. The proposed unsupervised clustering method is built on a novel amalgamation of variational autoencoder models with Bernoulli mixture models. To facilitate the training using gradient descent, we also introduce the Gumbel-Softmax distribution to resolve the infeasibility of the back-propagation algorithm in assessing categorical samples. As a primary application, the model is trained on image datasets to show performance compared to other state-of-the-art clustering algorithms (in terms of accuracy and compression rate). To the best of our knowledge, what we present is the first methodology for simultaneous data compression and unsupervised learning (in the compressed domain).

1 Introduction

Unsupervised clustering is a fundamental task underlying numerous real-world problems in machine learning, medical imaging, social network analysis, bioinformatics, computer graphics, etc. Performing clustering on compressed data can be treated as a specific solution to image clustering problems to meet the demands of the unstructured and unlabelled image collection. It can search the interested groups among compressed images, and moreover detect the scene changes of compressed videos to save the cost of transmission over the network or internet. The clustering capabilities of Nonnegative matrix factorization (NMF) was studied to learn parts of faces and semantic features of text Lee & Seung (1999), but it might fail to distinguish correlated clusters for image data Lazar & Doncescu (2009). The well-known model-based clustering methods, such as Gaussian mixture models (GMM), can represent the data population by estimating the corresponding distribution conditioning on different classes. However, the optimization of likelihood for mixture models relies on the Expectation Maximization (EM) algorithm, which has an inconsistent success rate and suffers computational inefficiency on high dimensional settings. To increase the accuracy and reduce the running time, it is reasonable to make inference on the extracted features of the raw pixels of images. It has been shown by recent works that Deep Neural Networks (DNNs) can learn nonlinear mappings and clustering-friendly representations to capture the distribution of image data (Xie et al. (2015), Song et al. (2013), Yang et al. (2016), Li et al. (2017), Yang et al. (2019)). Most

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

of these works jointly optimize both feature extraction and clustering to achieve comparable results.

Research shows that autoencoders can represent data via latent variables in a lower dimensional space. Autoencoding is done by iteratively learning two procedures: the encoder — non-linear mapping that transforms input data to a simpler code, and the decoder — reconstructing process that reproduce the original data. The variational autoencoder (VAE) behaves same and is able to learn the probabilistic models of data representation by the Auto-Encoding Variational Bayes (AEVB) algorithm, which is proposed to perform efficient posterior inference in directed probabilistic models for intractable posterior distributions and large datasets Kingma & Welling (2013). The algorithm introduces the Stochastic Gradient Variational Bayes (SGVB) estimator to optimize a recognition model that efficiently approximate posterior distribution without the need of expensive iterative inference schemes, such as Markov chain Monte Carlo (MCMC) methods. There are discussions about similarities and differences between training a variational autoencoder to minimize the reconstruction error and training the same autoencoder to compress the data (Ollivier (2014), Ballé et al. (2016)). This research shows that a autoencoder can be trained to minimize a tight upper bound on the compressed size (code length) of the data and some autoencoder-based compression methods are formally equivalent to VAEs. Furthermore, the flexible architecture of VAEs has potential to address an increasing need for lossy image compression, which is necessary for new or old image format devices such as unmanned aerial vehicles, mobile phones or remote sensors Theis et al. (2017).

To group a set of unlabelled objects after image compression, we propose a new approach that enable learning the binary representation of massive data and performing inference on this low dimensional domain. We achieve this using efficient neural network architectures that allow clustering of large images even on low-powered consumer devices. The model is trained in two steps: Firstly, a VAE is jointly trained with a Bernoulli mixture model (BMM) by the loss function consisting of a autoencoder reconstruction loss and a specific clustering loss. Then, the classifier will be updated after sampling of the posterior distribution, which is learned by the trained encoder. It is challenging to learn the discrete representation with neural network architectures because of the inability to backpropagate through non-differentiable samples. In our work, we use the Gumbel-Softmax Jang et al. (2016), which provides a differentiable sampling mechanism that trains the neural network with categorical reparameterization trick through estimating the gradient of the network parameters. Finally, our model can be decomposed into two parts: a data vendor that encodes the raw data into compressed data, and a data consumer that classifies the transmitted bit streams into classes as shown in Figure 1. The data vendor benefits from the communication

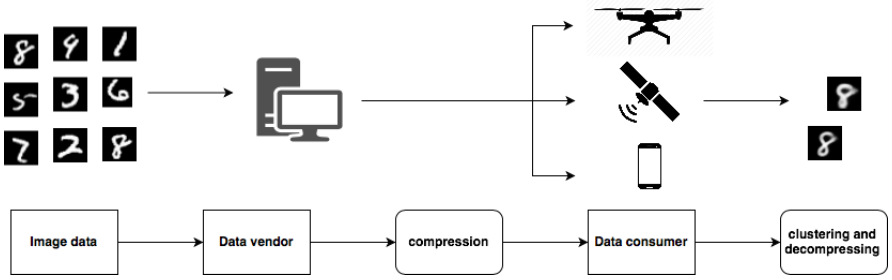


Figure 1: Compression and post-compression clustering

bandwidth because of the low bit-rate binary representations, and the data consumer benefits from low computational complexity by the efficiency of the neural network.

2 Related Work

In this section, we review some existing unsupervised clustering techniques and variational image compression works based on deep neural networks.

2.1 Unsupervised clustering

The classic clustering methods, such as K-means, Spectral clustering and DBSCAN, are fast and applicable to a wide range of problems. However, it is observed that these distance-based or similarity-based methods may perform poorly as the dimension of the analyzed data increases Steinbach et al. (2004). A number of works have shown that clusters for high dimensional data lie only in subsets of the full space and good data representations are beneficial to clustering. However, those feature extraction approaches based on linear methods may be inappropriate when different clusters lie in different subspaces. Deep Embedded Clustering (DEC) was proposed to jointly learn feature representations and assign clusters using a class of feedforward artifical neural networks Xie et al. (2015). It achieves impressive performances on clustering tasks and it is often used as a baseline for new DNN-based clustering methods. Following a similar motivation, in Yang et al. (2016) the authors learned Joint Unsupervised LEarning (JULE) to combine agglomerative clustering with the Convolutional Neural Network (CNN) and formulated them as a recurrent process. In Li et al. (2017), an end-to-end image representations method named fully convolutional autoencoder (FCAE) was proposed, and a discriminatively boosted clustering (DBC) framework was learned based on the FCAE. DBC highlightes high score assignments and deemphasize low score ones. With this soft clustering method, the autoencoder is more discriminative for latter clustering assignments and enlarges cluster purities. In Yang et al. (2019), an identification criterion was proposed to address the identifiability issues for nonlinear mixture models and was implemented by a neural network. Although all of these works are able learn the non-linear mappings specifically for clustering-friendly representations, they fail to expose a connection between directed probabilistic models and representations of the data. Variational Deep Embedding (VaDE) Jiang et al. (2017) applies a mixture of Gaussian models as the prior distribution to replace the prior distribution of standard VAE, and therefore enable modelling the data generative procedure jointly with GMM and DNNs. In contrast to this method, we make inference on binary data representations with the framework Gumbel-softmax estimator proposed by Jang et al. (2016), and discuss the potential of our model for compressed image.

2.2 Image compression

Image compression is a type of data compression and might be lossy or lossless. Lossless compression provides an error-free way to compress data and reconstitute it into its original state, while lossy compression does not preserve the information completely but achieves a high compression ratio. The autoencoder was applied for lossy image compression and achieved comparable results in recent works (Zhou et al. (2018), Theis et al. (2017), Ballé et al. (2018)). Motivated by the potential of autoencoders to address the need of more flexible compression algorithms, the authors in Theis et al. (2017) proposed a new approach to optimize autoencoders for lossy image compression. Ballé et al. (2018) described an end-to-end trainable model for image compression based on VAE and this model can effectively capture spatial dependencies in the latent representations. In Zhou et al. (2018), the authors also presented an end-to-end image compression framework for low bit-rate image compression based on VAE. Its prior probability of compressed representation was modeled by a Laplacian distribution and an effective convolution-based post-processing module was proposed to remove the compression noise for low bit-rate images. Taken together, prior research provides evidence that a better model fit leads to a better compression performance, and consequently enable a more accurate clustering assignment.

3 Methods

Considering the dataset $\mathbf{x}$ with N identically independently distributed (i.i.d) samples $\{x_i\}_{i=1}^N$ and $x_i \in \mathbf{R}^d$, we assume that the data are generated by some random process, involving an unobserved Bernoulli random variable z which belongs to one of k classes. We will firstly introduce the model and then describe how to use deep learning architecture to optimize the object.

3.1 The generative model

We begin with the joint distribution $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}, \mathrm{c}) = p_{\boldsymbol{\theta}}(\mathrm{c}) p_{\boldsymbol{\theta}}(\mathbf{z}|\mathrm{c}) p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}, \mathrm{c})$, where $\boldsymbol{\theta}$ is denoted as the generative model parameters and learned by DNNs as described in section 3.4. It says that an

observe $\mathbf{x}$ is generated from a set of latent variables $\mathbf{z}$, and $\mathbf{z}$ follows the mixture distributions with respect to ($w.r.t.$) the classes variable c. Note that $\mathbf{z}$ here is binarized and distributed varying with classes, while it are considered as continuous and standard normal vector in other general variational models. Their distributions are described as:

$$\mathbf{x} \sim Bernoulli(\boldsymbol{\mu_x}) \; or \; \mathbf{x} \sim N(\boldsymbol{\mu_x}, \boldsymbol{\sigma_x}\mathbf{I}) \tag{1}$$

$$\mathbf{z} \sim Bernoulli(\boldsymbol{\mu_z}) \tag{2}$$

$$\mathrm{c} \sim Categorical(\boldsymbol{\pi}) \tag{3}$$

Along with this generative process, we assume

$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}, \mathrm{c}) = p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$$

which means that $\mathbf{x}$ and c are independent conditioning on $\mathbf{z}$. As the same setting in Kingma & Welling (2013), we denote the recognition model $q_{\boldsymbol{\phi}}(\mathbf{z}, \mathrm{c}|\mathbf{x})$ as the variational approximation under the KL-divergence to the intractable posterior $p_{\boldsymbol{\theta}}(\mathbf{z}, \mathrm{c}|\mathbf{x})$ and $\boldsymbol{\phi}$ is the recongnition model parameters. To learn the recognition model parameters $\boldsymbol{\phi}$ jointly with the generative model parameters $\boldsymbol{\theta}$ by optimizing the likelihood through gradient descent in stochastic neural networks, we replace the non-differentiable categorical sample $\boldsymbol{z}$ with the Gumbel-Softmax estimator $\boldsymbol{y}$ and discretize $\boldsymbol{y}$ as binary sequence using $\arg\max$. Therefore we use the discrete code as clustering input but use our continuous approximation for the back pass by approximating $\nabla_\theta \mathbf{z}$ with$\nabla_\theta \mathbf{y}$. The details are described in 3.4.

3.2 The variational lower bound

The loglikelihood of the whole observed data $\mathbf{x}$ is $\log p(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(N)}) = \sum_{i=1}^{N} \log p(\mathbf{x}^{(i)})$. Each likelihood element is written as:

$$\log p(\mathbf{x}^{(i)}) = D_{\mathrm{KL}}(q_\phi(\mathbf{z}, \mathrm{c}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z}, \mathrm{c}|\mathbf{x}^{(i)})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) \tag{4}$$

The first RHS term is the KL-divergence of the approximation $q_\phi(\mathbf{z}, \mathrm{c}|\mathbf{x}^{(i)})$ from the true posterior $p_{\boldsymbol{\theta}}(\mathbf{z}, \mathrm{c}|\mathbf{x}^{(i)})$ and

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = E_{q_\phi(\mathbf{z}, \mathrm{c}|\mathbf{x}^{(i)})}\left[\log p_\theta(\mathbf{x}^{(i)}, \mathbf{z}, \mathrm{c}) - \log q_\phi(\mathbf{z}, \mathrm{c}|\mathbf{x}^{(i)})\right] \tag{5}$$

Since the KL-divergence is non-negative and the value of $\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$ is not related with $\boldsymbol{\phi}$, minimizing the KL-divergence is the same as maximizing the evidence lower bound. It is then able to find the parameters that gives as tight a bound as possible on the marginal probability of $\mathbf{x}$.

3.3 The reparameterization trick with Gumbel-softmax estimators

Applying SGVB estimator in the following two steps Kingma & Welling (2013):
Step 1). reparameterizing the random variable $\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})$ with a differentiable transformation $g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x}^{(i)})$ of an noise variable $\boldsymbol{\epsilon}$:

$$\mathbf{z} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x}^{(i)}) \quad \text{where} \quad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$$

Step 2). forming the estimator expectations of some function $h(\mathbf{z})$ $w.r.t.$ $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})$ as follows:

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}[h(\mathbf{z})] = \mathbb{E}_{p(\epsilon)}\left[h\left(g_\phi\left(\boldsymbol{\epsilon}, \mathbf{x}^{(i)}\right)\right)\right] = \frac{1}{L}\sum_{l=1}^{L} f\left(g_\phi\left(\boldsymbol{\epsilon}^{(l)}, \mathbf{x}^{(i)}\right)\right) + o_p(1) \tag{6}$$

where $\boldsymbol{\epsilon}^{(l)}$ are samples generated from $p(\boldsymbol{\epsilon})$. To utilize this reparameterization trick for non-differentiable discrete $\mathbf{z}$, we replace categorical samples with Gumbel Softmax samples as introduced in Jang et al. (2016). The mapping corresponding with the noise $\boldsymbol{\epsilon}$ is found as:

$$y_i = \frac{\exp\left(\left(\log\left(\mu_i\right) + \epsilon_i\right)/\tau\right)}{\sum_{j=1}^{k}\exp\left(\left(\log\left(\mu_j\right) + \epsilon_j\right)/\tau\right)} \quad \text{for } i = 1, \ldots, k \tag{7}$$

where $\epsilon_1 ... \epsilon_k$ are i.i.d samples drawn from Gumbel (0,1) distribution, μ_i are the probability of belonging to classes i and τ is the softmax temperature. It has been shown that samples $\boldsymbol{y}$ will become

one-hot and the Gumbel-Softmax distribution will converge to the distribution of samples $\boldsymbol{z}$ Jang et al. (2016).

$$\boldsymbol{z} \rightarrow \boldsymbol{y} \quad \text{as} \quad \tau \rightarrow 0 \tag{8}$$

From deterministic mappings in (7) and (8), we have the approximation

$$p(\boldsymbol{\epsilon}) \prod_i d\epsilon_i = q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{x}) \prod_i d\mathrm{y}_i \rightarrow q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \prod_i d\mathrm{z}_i \quad \text{as} \quad \tau \rightarrow 0$$

and

$$\int q_\phi(\mathbf{y}|\mathbf{x}) f(\mathbf{y}) d\mathbf{y} = \int p(\boldsymbol{\epsilon}) f\left(g_\phi(\boldsymbol{\epsilon}, \mathbf{x})\right) d\boldsymbol{\epsilon} = \frac{1}{L} \sum_{l=1}^{L} f\left(g_\phi\left(\boldsymbol{\epsilon}^{(l)}, \mathbf{x}\right)\right) + o_p(1) \tag{9}$$

3.4 Clustering with variational models

We want to differentiate and optimize the lower bound $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$ $w.r.t.$ both the recognition model parameters $\boldsymbol{\phi}$ and generative parameters $\boldsymbol{\theta}$. Rewrite the equation (5) as,

$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = E_{q_{\boldsymbol{\phi}}(\mathbf{z},\mathrm{c}|\mathbf{x}^{(i)})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{z}|\mathrm{c}) + \log p_{\boldsymbol{\theta}}(\mathrm{c}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)}) - \log q_{\boldsymbol{\phi}}(\mathrm{c}|\mathbf{z})]$

The RHS of above equation is the sum of following four terms

$$A_1 = E_{q_{\boldsymbol{\phi}}(\mathbf{z},\mathrm{c}|\mathbf{x}^{(i)})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z})\right] \tag{10}$$

$$A_2 = -E_{q_{\boldsymbol{\phi}}(\mathbf{z},\mathrm{c}|\mathbf{x}^{(i)})}\left[\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})\right] \tag{11}$$

$$A_3 = E_{q_{\boldsymbol{\phi}}(\mathbf{z},\mathrm{c}|\mathbf{x}^{(i)})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{z}|\mathrm{c})\right] \tag{12}$$

$$A_4 = E_{q_{\boldsymbol{\phi}}(\mathbf{z},\mathrm{c}|\mathbf{x}^{(i)})}\left[\log p_{\boldsymbol{\theta}}(\mathrm{c}) - \log q_{\boldsymbol{\phi}}(\mathrm{c}|\mathbf{z})\right] \tag{13}$$

Apply the approximation (9) and we estimate equation $A_1 - A_4$ as following

$$\hat{A}_1 = \frac{1}{L} \sum_{l=1}^{L} \log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}|\boldsymbol{y}^{(i,l)}\right) \tag{14}$$

$$\hat{A}_2 = -\boldsymbol{\mu}_{\mathbf{z}}^{(i)} \log(\boldsymbol{\mu}_{\mathbf{z}}^{(i)}) - (1 - \boldsymbol{\mu}_{\mathbf{z}}^{(i)}) \log(1 - \boldsymbol{\mu}_{\mathbf{z}}^{(i)}) \tag{15}$$

$$\hat{A}_3 = \sum_{\mathrm{c}=1}^{k} q_{\boldsymbol{\phi}}(\mathrm{c}|\boldsymbol{y}^{(i,l)}) \frac{1}{L} \sum_{l=1}^{L} \log p_{\boldsymbol{\theta}}(\boldsymbol{y}^{(i,l)}|\mathrm{c}) \tag{16}$$

$$\hat{A}_4 = \sum_{c=1}^{k} \frac{1}{L} \sum_{l=1}^{L} q_{\boldsymbol{\phi}}(\mathrm{c}|\mathbf{y}^{(i,l)}) \left[\log p_{\boldsymbol{\theta}}(\mathrm{c}) - \log q_{\boldsymbol{\phi}}(\mathrm{c}|\mathbf{y}^{(i,l)})\right] \tag{17}$$

where Gumbel Softmax samples $\boldsymbol{y}^{(i,l)} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(l)}, \mathbf{x}^{(i)})$ and $\epsilon_i^{(l)} \sim G(0, 1)$. By optimizing equation 5, we will learn $p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\boldsymbol{y}^{(i,l)})$ by the decoder neutral network $f(\mathbf{x}; \boldsymbol{\theta})$ and learn the posterior parameters $\boldsymbol{\mu}_z^{(i)}$ by the encoder neutral network $g(\mathbf{x}; \boldsymbol{\phi})$. For each generated sample $\boldsymbol{y}^{(i,l)}$ corresponding with each input $\mathbf{x}^{(i)}$, we update the classifiers by,

$$q_{\boldsymbol{\phi}}(\mathrm{c}|\boldsymbol{y}^{(i,l)}) = \frac{p_{\boldsymbol{\theta}}(\mathrm{c}) p_{\boldsymbol{\theta}}(\boldsymbol{y}^{(i,l)}|\mathrm{c})}{\sum_{c=1}^{k} p_{\boldsymbol{\theta}}(\mathrm{c}) p_{\boldsymbol{\theta}}(\boldsymbol{y}^{(i,l)}|\mathrm{c})}$$

and the parameters $\boldsymbol{\pi}$ in $p_{\boldsymbol{\theta}}(c)$ and $\boldsymbol{\mu}_z^{(\mathrm{c})}$ in $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathrm{c})$ is setted as the model parameters. The diagram of this process is illustrated in Figure 2. We are finally able to construct an estimator of the marginal likelihood lower bound of the full N sample data set based on mini batches:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = \widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) + o_p(1) = \frac{N}{M} \sum_{i=1}^{M} \widetilde{\mathcal{L}}\left(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}\right) + o_p(1) \tag{18}$$

where the mini batch $\mathbf{x}^M = \mathbf{x}^{(i)}{}_{i=1}^{M}$ is a randomly drawn sample of M data points from the full data set X. It is pointed that the number of samples L per data point can be set to 1 as long as the mini batch size M was large enough, e.g. M = 100 Kingma & Welling (2013). Derivatives $\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \widetilde{\mathcal{L}}\left(\boldsymbol{\theta}; \mathbf{x}^M\right)$ can be taken, and we use Adam Kingma & Ba (2014) to optimize the approximated lower bound, which calculates an exponential moving average of the gradient and the squared gradient with two parameters control the decay rates of these moving averages.

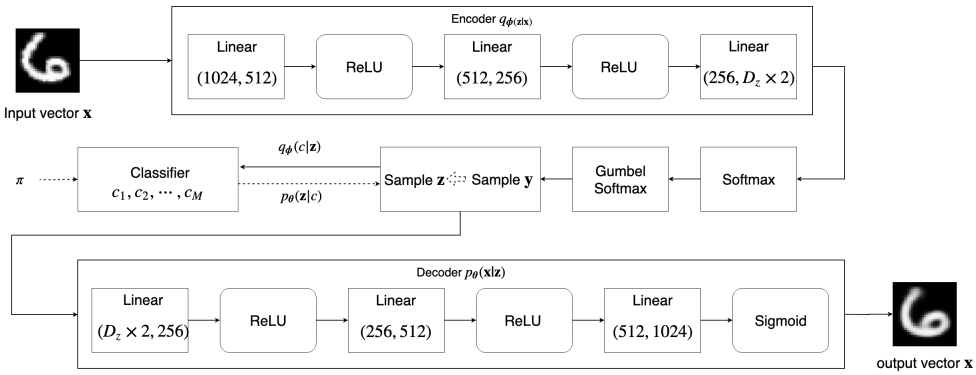


Figure 2: The diagram of clustering analysis with the variational autoencoder architecture.

4 Experiments

4.1 Evaluation Metric

For clustering performance, we follow the same evaluation metric mentioned in Xie et al. (2015) and Jiang et al. (2017) to make fair comparison. With given number of clusters, we define unsupervised clustering accuracy (ACC) as:

$$ACC = \max_{m \in \mathcal{M}} \frac{\sum_{i=1}^{N} I\{l_i = m(c_i)\}}{N}$$

where N is the total number of samples, l_i is the ground-truth label, c_i is the cluster assignment obtained by the model and $\mathcal{M}$ ranges over all possible mappings between predicted labels and the true labels. The range of ACC value is in $[0, 1]$ and higher ACC value indicates more accurate clustering performance.

In terms of the compression quality, we use the peak signal-to-noise ratio (PSNR) between images to compare the distance of reconstructed image with the original image. The higher the PSNR, the better the quality of the reconstruction. It is noted that acceptable PSNR for wireless transmission is from 20 dB to 25 dB Thomos et al. (2006). We will evaluate both clustering performance and the compression quality across the value of bits per pixel (bpp)— the number of bits of information stored per pixel. The more bits there are, the more colours can be represented, but the more memory is required to store or display the image.

4.2 Experiment Setups

We present the performance of our method compared with the others on the hand-written digit image dataset MNIST. This dataset is composed of centered 28×28 gray-scale images and contains 70000 image samples (60000 train image data and 10000 test image data) LeCun & Cortes (2010). We resize the image as 32 when load the dataset. To make the performance convincing and applicable in general, we train the model on the train set (60000 data) and then test the performance of best trained model on the test set (10000 data). The total number of classes $K = 10$ is assigned as known before the procedure.

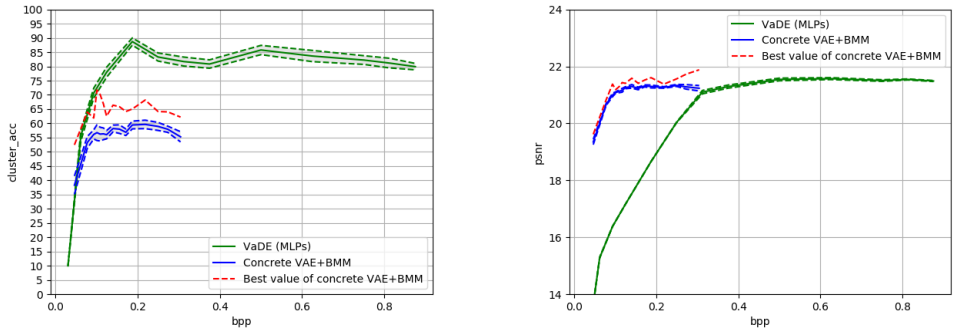
For the neural work architecture, we use the multilayer perceptron (MLP), a class of feedforward artificial neural networks. The encoder neutral network $g(\mathbf{x}; \boldsymbol{\phi})$ with layers $D_z - 512 - 256 - 1024$ is to learn the recognition model parameters $\boldsymbol{\phi}$, and the decoder neutral network $f(\mathbf{x}; \boldsymbol{\theta})$ with layers $1024 - 512 - 256 - D_z$ is to learn the generative model parameters $\boldsymbol{\theta}$. We denote D_z here as the dimension of the latent variable $\mathbf{z}$ of our model. The value of bpp is changed corresponding to the value of D_z. For example, with $D_z = 28$, one gray-scale image input will generate (1, 28) binary codes after compressed, then we will have $28/1024 = 0.02734375$ bpp in the further compression step. All layers are fully connected and followed with a Rectified linear unit (ReLU) — a nonlinear activation function defined as the positive part of its argument.

For training, we used Adam Kingma & Ba (2014), a variant of stochastic gradient descent, to jointly optimize the full set of parameters with $\beta = (0.9, 0.999)$. The learning rate is initialized as 0.001 and decreases every 10 epochs with a decay rate of 0.9 down to the minimum of 0.0002. We report all results averaged from 10 experiments across different bpp value.

We compare our approach with K-means, GMM and the VaDE framework Jiang et al. (2017). We apply K-means and GMM directly on the raw image pixels with default settings. The VaDE is designed to generate and group the image data through modelling the continuous latent variable. It utilizes a similar MLP architecture as DEC with the encoder $1024 - 500 - 500 - 2000 - 10$ and the decoder $10 - 2000 - 500 - 500 - 1024$. It is shown in Jiang et al. (2017) that VaDE achieved promising clustering accuracy compared with DEC and other methods. We report the results of VaDE by re-running the code released by the original paper. The results we obtain are somewhat different from the one reported because of different experimental settings and the random seeds controlling.

4.3 Experiment Results

Figure 3a and Figure 3b compares the clustering performance and the compression quality against bpp respectively. The solid line represents the mean of 10 runs under each setting, while filling grey area represents the expected errors of the mean in 10 runs. It can be seen that our method can achieve comparable clustering accuracy at the very low bit rates. In the meanwhile, the compression quality outperform the VaDE framework as shown in Figure 3b. Table 1 presents the best clustering accuracy of our method with other baselines.



(a) Clustering accuracy (%) against BPP

(b) PSNR against BPP

Figure 3: Experiment results averaged from 10 experiments are presented by the solid line against bpp values. The grey area between two dish lines shows the standard deviations of the mean in this 10 runs.

Method	K-means	GMM	VaDE	Concrete VAE+BMM
Best Clustering Acuraccy (%)	55.37	42.22	95.30	71.69

Table 1: Best clustering accuracy (%) performance comparison on MNIST test data

5 Discussion

In this paper, we proposed a new methodology that enables accessing the target image data easily and quickly by addressing the image clustering problem with image compression structure. The method is presented as a novel amalgamation of the variational autoencoder (VAE) with the Bernoulli mixture model (BMM). The VAE performs the flexible encoding and decoding of images, and the BMM provides the model-based clustering for binary data. In details, we utilized a deep learning architecture to train this model through the SGVB estimator and the reparameterization

trick, and optimize it with a two-fold loss function consisting of a reconstruction loss and a clustering loss. Specifically, we apply the Gumbel-Softmax distribution to address the issues caused by the non-differentiable samples. Finally, the best model can achieve a good improvement in clustering accuracy over the well-known clustering methods, such as K-means and GMM.

Our method is build on two previous publications (Jiang et al. (2017), Dilokthanakul et al. (2016)) but we focused on the clustering analysis for binary data. Both of the previous research proposed image generative procedures through VAE framework with the assumption that the continuous latent variable $\mathbf{z}$ is normally distributed. In this paper, we focused attention to a mixture model approach for binary data, which requires fewer storage space and has the potential to reduce the transmission bandwidth. It seems that the previous VaDE framework has a similar potential compression ability as our method in Figure 3b. However, its effectiveness for compression may be severely compromised because of the fundamental differences between generative models and compression framework lead by the role of discretization in lossy compression systems. On the other hand, in our method, an approximate mixture of discrete probability models is provided so discrete entropy can be reasonably equated to the actual rate of a discrete code. We conclude that representing the image data with binary code, our method still has a better potential than the VaDE framework, and we do not need a further discretizing process for compression purpose. It worthy noting that the compression ability of our method is not comprarable with other wee-developed compression methods in Figure 3b,

We applied modern Bayesian approaches to clustering the multivariate binary data. Bayesian methods benefits the generality and computational power. However, because the variance highly related to the success mean in Bernoulli mixture model, it is much challenging to distinguish different classes. We highlights that it is important to develop more advanced applications and tools in this field with the increasing prevalance of high-dimensional binary data.

References

Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end optimized image compression. *CoRR*, abs/1611.01704, 2016. URL `http://arxiv.org/abs/1611.01704`.

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior, 2018.

Nat Dilokthanakul, Pedro A. M. Mediano, Marta Garnelo, Matthew C. H. Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders, 2016.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax, 2016.

Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: an unsupervised and generative approach to clustering. pp. 1965–1972, 2017.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL `http://arxiv.org/abs/1412.6980`.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

C. Lazar and A. Doncescu. Non negative matrix factorization clustering capabilities; application on multivariate image segmentation. In *2009 International Conference on Complex, Intelligent and Software Intensive Systems*, pp. 924–929, March 2009. doi: 10.1109/CISIS.2009.190.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999. doi: 10.1038/44565. URL `https://doi.org/10.1038/44565`.

Fengfu Li, Hong Qiao, Bo Zhang, and Xuanyang Xi. Discriminatively boosted image clustering with fully convolutional auto-encoders, 2017.

Yann Ollivier. Auto-encoders: reconstruction versus compression. *CoRR*, abs/1403.7752, 2014.

Chunfeng Song, Feng Liu, Yongzhen Huang, Liang Wang, and Tieniu Tan. Auto-encoder based data clustering. In José Ruiz-Shulcloper and Gabriella Sanniti di Baja (eds.), *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 117–124, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-41822-8.

Michael Steinbach, Levent Ertöz, and Vipin Kumar. *The Challenges of Clustering High Dimensional Data*, pp. 273–309. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-662-08968-2. doi: 10.1007/978-3-662-08968-2_16. URL https://doi.org/10.1007/978-3-662-08968-2_16.

Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszr. Lossy image compression with compressive autoencoders, 2017.

Nikolaos Thomos, Nikolaos V Boulgouris, and Michael G Strintzis. Optimized transmission of jpeg2000 streams over wireless channels. *Image Processing, IEEE Transactions on*, 15:54 – 67, 02 2006. doi: 10.1109/TIP.2005.860338.

Junyuan Xie, Ross B. Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. *CoRR*, abs/1511.06335, 2015. URL http://arxiv.org/abs/1511.06335.

Bo Yang, Xiao Fu, Nicholas D. Sidiropoulos, and Kejun Huang. Learning nonlinear mixtures: Identifiability and algorithm. *CoRR*, abs/1901.01568, 2019. URL http://arxiv.org/abs/1901.01568.

Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters, 2016.

Lei Zhou, Chunlei Cai, Yueming Gao, Sanbao Su, and Junmin Wu. Variational autoencoder for low bit-rate image compression. In *CVPR Workshops*, 2018.