Unsupervised Clustering Using a Variational Autoencoder with Constrained Mixtures for Posterior and Prior

Mashfiqul Huq Chowdhury[0000-0003-3438-6645], Yuichi Hirose[0000-0001-5565-3314], Stephen Marsland[0000-0002-9568-848X], and Yuan Yao[0009-0002-2381-0033]

Victoria University of Wellington, Wellington, New Zealand {mashfiq.chowdhury, yuichi.hirose, stephen.marsland, yuan.yao}@vuw.ac.nz

1 Estimation of $q(y_k)$

We need to solve the following Lagrangian function with the Lagrange multiplier λ:

$$\frac{1}{L}\sum_{k=1}^{K} q(y_k)\sum_{l=1}^{L}\log\left[\frac{q_{\boldsymbol{\phi}}(\mathbf{z}_k^{(l)}|\mathbf{x}, y_k)\ q(y_k)}{p(\mathbf{z}_k^{(l)}|y_k)\ p(y_k)}\right] + \lambda\left[1-\sum_{k=1}^{K} q(y_k)\right]. \tag{1}$$

Differentiating (1) with respect to $q(y_k)$ and setting the derivative equal to 0 gives:

$$\frac{1}{L}\sum_{l=1}^{L}\log\left[\frac{p(\mathbf{z}_k^{(l)}|y_k)\ p(y_k)}{q_{\boldsymbol{\phi}}(\mathbf{z}_k^{(l)}|\mathbf{x},\ y_k)\ q(y_k)}\right] = 1-\lambda. \tag{2}$$

Multiplying both sides by L and exponentiating both sides gives:

$$\prod_{l=1}^{L}\left[\frac{p(\mathbf{z}_k^{(l)}|y_k)\ p(y_k)}{q_{\boldsymbol{\phi}}(\mathbf{z}_k^{(l)}|\mathbf{x}, y_k)\ q(y_k)}\right] = \exp(L-\lambda L). \tag{3}$$

Multiplying $q(y_k)$ on both sides and summing over $k = 1$ to K with the constraint $\sum\limits_{k=1}^{K} q(y_k) = 1$ to get:

$$\sum_{k=1}^{K}\prod_{l=1}^{L}\left[\frac{p(\mathbf{z}_k^{(l)}|y_k)\ p(y_k)}{q_{\boldsymbol{\phi}}(\mathbf{z}_k^{(l)}|\mathbf{x},\ y_k)}\right] = \exp(L-\lambda L). \tag{4}$$

Substituting (4) into (3), we obtain a closed-form expression of the $q(y_k)$:

$$\hat{q}(y_k) = \frac{\prod\limits_{l=1}^{L}\left[\frac{p(y_k)\ p(\mathbf{z}_k^{(l)}|y_k)}{q_{\boldsymbol{\phi}}(\mathbf{z}_k^{(l)}|\mathbf{x},\ y_k)}\right]}{\sum\limits_{j=1}^{K}\prod\limits_{l=1}^{L}\left[\frac{p(y_j)\ p(\mathbf{z}_j^{(l)}|y_j)}{q_{\boldsymbol{\phi}}(\mathbf{z}_j^{(l)}|\mathbf{x},\ y_j)}\right]}. \tag{5}$$

2 Training Algorithm

Algorithm 1: Training of the β-MVAE (EM) and MVAE (EM) models.

```
Input: $\mathcal{D}_{\text{Train}} = \{\mathbf{x}_i\}_{i=1}^{N}$; $\mathbf{x}_i \in \mathbb{R}^D$ or $\mathbb{Z}^D$, number of clusters: $K$, Latent
       variables: $\mathbf{z} \in \mathbb{R}^M$, $\mathbf{y} = \{y_1, \cdots, y_K\}$, regularization coefficient:
       $\beta > 0$ ($\beta = 1$: MVAE(EM) model), learning rate: $\eta$, mini-batch
       size: $B$, Monte Carlo sample size: $L$, number of epochs: $E$.
Output: $\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\psi}$
Initialize $\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\psi}$.
repeat
    for each epoch $e = 1, \cdots, E$ do
        randomly select $\mathbf{x}_i \in \mathcal{B} = \{\mathbf{x}_1, \cdots, \mathbf{x}_B\}$ from $\mathcal{D}_{\text{Train}}$
        for each $k \in \{1, \cdots, K\}$ do
            $(\boldsymbol{\mu}_{\boldsymbol{\phi}k}(\mathbf{x}_i), \log \boldsymbol{\sigma}^2_{\boldsymbol{\phi}k}(\mathbf{x}_i)) = \text{Encoder NN}_{\boldsymbol{\phi}k}(\mathbf{x}_i)$.
            for $l = 1, \cdots, L$ do
                $$\mathbf{z}_{ik}^{(l)} \leftarrow \boldsymbol{\mu}_{\boldsymbol{\phi}k}(\mathbf{x}_i) + \boldsymbol{\sigma}_{\boldsymbol{\phi}k}(\mathbf{x}_i) \odot \boldsymbol{\epsilon}^{(l)};\ \boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon}) = \mathcal{N}(\mathbf{0}, \mathbf{I}).$$
            Compute $\hat{q}(y_{ik})$ using (5)
            Compute:
            $$\tilde{l}_{ik} = \frac{1}{L} \sum_{l=1}^{L} \left[ \log p_\theta(\mathbf{x}_i | \mathbf{z}_{ik}^{(l)}) - \beta \left\{ \log q_\phi(\mathbf{z}_{ik}^{(l)} | \mathbf{x}_i, y_{ik}) - \log p(\mathbf{z}_{ik}^{(l)} | y_{ik}) - \log p(y_{ik}) \right\} \right]$$
        Compute:
            $$\tilde{s}_i = \sum_{k=1}^{K} \hat{q}(y_{ik}) [\tilde{l}_{ik} - \beta \log \hat{q}(y_{ik})].$$
            $$\tilde{h}_i = \tilde{l}_{i\hat{k}};\quad \hat{k} = \arg\max_k \hat{q}(y_{ik}).$$
        Soft optimization ELBO: $\tilde{\mathcal{L}}_{\boldsymbol{\phi},\boldsymbol{\theta},\boldsymbol{\psi},\hat{q}(y)}(\mathcal{B}) = \sum_{i=1}^{B} \tilde{s}_i$.
        Hard optimization ELBO: $\tilde{\mathcal{L}}_{\boldsymbol{\phi},\boldsymbol{\theta},\boldsymbol{\psi},\hat{q}(y)}(\mathcal{B}) = \sum_{i=1}^{B} \tilde{h}_i$.
        Update parameters:
        $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} - \eta \frac{1}{B} \bigtriangledown_{\boldsymbol{\phi}} \tilde{\mathcal{L}}_{\boldsymbol{\phi},\boldsymbol{\theta},\boldsymbol{\psi},\hat{q}(y)}(\mathcal{B})$.
        $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \frac{1}{B} \bigtriangledown_{\boldsymbol{\theta}} \tilde{\mathcal{L}}_{\boldsymbol{\phi},\boldsymbol{\theta},\boldsymbol{\psi},\hat{q}(y)}(\mathcal{B})$.
        $\boldsymbol{\psi} \leftarrow \boldsymbol{\psi} - \eta \frac{1}{B} \bigtriangledown_{\boldsymbol{\psi}} \tilde{\mathcal{L}}_{\boldsymbol{\phi},\boldsymbol{\theta},\boldsymbol{\psi},\hat{q}(y)}(\mathcal{B})$.
        return $\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\psi}$.
until number of iterations completed.
```

3 Clustering Procedures

We train each evaluated model-VAE [3] , VADE [2], k-DVAE [1], MVAE(EM), and β-MVAE(EM)-with five different random initializations. The experimental setup is detailed in the main article.

3.1 Comparator Algorithms

We train the VAE, VADE, and k-DVAE algorithms, then use the trained model to extract the latent representations from the test set of each dataset. Next, we apply the Gaussian mixture model (GMM) to the learned latent embeddings for clustering and report the highest unsupervised clustering accuracy achieved by each model across datasets, based on results from five experiments.

3.2 Proposed Algorithms

We train our proposed models, including MVAE(EM), and β-MVAE(EM), using both soft and hard optimization approaches. The trained model is then applied to the test set of each benchmark dataset, generating K latent representations for each observation. We select the latent representation based on the MAP estimate of the clustering assignment probabilities using (5). Next, we apply the GMM over the learned latent embeddings for clustering and report the highest unsupervised clustering accuracy achieved by the proposed models across datasets, based on results from five experiments.

The next section reports the latent dimension (M) of each evaluated model.

4 Latent Dimensions

4.1 VAE

We achieved the best clustering performance with $M = 5$ for the Fashion-MNIST, Digits, HAR, and USPS datasets and with $M = 10$ for the MNIST, STL-10, and Reuters datasets.

4.2 VADE

The VADE algorithm [2] uses the following equation to calculate clustering assignment probabilities:

$$\hat{q}(y_k) = \frac{p(y_k)\ p(\mathbf{z}|y_k)}{\sum\limits_{j=1}^{K} p(y_j)\ p(\mathbf{z}|y_j)}. \qquad (6)$$

Clustering assignment is predicted by using:

$$\hat{y} = \arg\max_{k} \hat{q}(y_k). \tag{7}$$

We applied both the posterior assignment estimate in (6) and the GMM for clustering, achieving the best performance using GMM. The optimal performance is obtained with $M = 5$ for the Fashion MNIST, Digits, HAR, and STL-10 datasets and with $M = 10$ for the MNIST, USPS, and Reuters datasets.

4.3 *k*-DVAE

We obtained the best clustering performance with $M = 5$ for the HAR, USPS, Reuters, Fashion-MNIST, and Digits datasets and with $M = 10$ for the MNIST and STL-10 datasets.

4.4 MVAE(EM)

We trained the MVAE(EM) algorithm using soft and hard optimization approaches. We achieved good clustering results with a latent dimension of $M = 5$ for the USPS; $M = 6$ Fashion-MNIST dataset; $M = 7$ for the MNIST and Reuters dataset; and $M = 10$ for the Digits, HAR, and STL-10 datasets under the soft optimization of the objective function. Again for the hard optimization approach, we obtained good clustering performance with a latent dimension of $M = 5$ for the STL-10, Fashion-MNIST, Reuters, and USPS datasets; and $M = 10$ for the MNIST, Digits, and HAR datasets.

4.5 β-MVAE(EM)

Like the MVAE(EM) model, we trained the MVAE(EM) algorithm using soft and hard optimization approaches. We achieved the best clustering performance with $M = 5$ for the USPS and Reuters datasets; and $M = 10$ for the Digits, MNIST, Fashion-MNIST, STL-10, and HAR datasets under the soft optimization of the objective function. Again for the hard optimization approach, we obtained the best clustering performance with $M = 5$ for Digits, Fashion-MNIST, Reuters, USPS, and HAR datasets; and with $M = 10$ for MNIST and STL-10 datasets.

References

1. Caciularu, A., and Goldberger, J. An entangled mixture of variational autoencoders approach to deep clustering. *Neurocomputing 529* (2023), 182–189.
2. Jiang, Z., Zheng, Y., Tan, H., Tang, B., and Zhou, H. Variational deep embedding: A generative approach to clustering. *CoRR, abs/1611.05148 1* (2016).
3. Kingma, D. P., and Welling, M. Auto-encoding variational Bayes. *International Conference on Learning Representations* (2013).