Pattern Recognition 107 (2020) 107514

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Dirichlet Variational Autoencoder

Weonyoung Joo[a], Wonsung Lee[b], Sungrae Park[c], Il-Chul Moon[a,*]

[a] Department of Industrial and Systems Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea
[b] AI Technology Unit, SK Telecom, Seoul 04539, Republic of Korea
[c] Clova AI Research, NAVER Corp., Gyeonggi-do 13561, Republic of Korea

ARTICLE INFO

Article history:
Received 23 April 2019
Revised 4 October 2019
Accepted 22 June 2020
Available online 24 June 2020

Keywords:
Representation learning
Variational autoencoder
Deep generative model
Multi-modal latent representation
Component collapse

ABSTRACT

This paper proposes Dirichlet Variational Autoencoder (DirVAE) using a Dirichlet prior. To infer the parameters of DirVAE, we utilize the stochastic gradient method by approximating the inverse cumulative distribution function of the Gamma distribution, which is a component of the Dirichlet distribution. This approximation on a new prior led an investigation on the component collapsing, and DirVAE revealed that the component collapsing originates from two problem sources: decoder weight collapsing and latent value collapsing. The experimental results show that 1) DirVAE generates the result with the best log-likelihood compared to the baselines; 2) DirVAE produces more interpretable latent values with no collapsing issues which the baselines suffer from; 3) the latent representation from DirVAE achieves the best classification accuracy in the (semi-)supervised classification tasks on MNIST, OMNIGLOT, COIL-20, SVHN, and CIFAR-10 compared to the baseline VAEs; and 4) the DirVAE augmented topic models show better performances in most cases.

1. Introduction

1.1. Motivations

A latent variable model (LVM) such as Latent Dirichlet Allocation (LDA) [1] brought huge success in the machine learning field. The key assumption in the LVMs is that there is a relatively simple latent manifold which preserves the property of original data in the low-dimensional space. In the perspective of dimensionality reduction of LVMs, Wong [2] developed a linear dimensionality reduction technique which aims to preserve the non-negative sparse reconstructive relationship of the data in the relatively low-dimensional space. Also, Lee et al. [3] suggested an approach which preserves the local intra-structure of different source of data by latent representation in common low-dimensional space. Meanwhile, a *Variational Autoencoder* (VAE) [4] led LVMs to remarkable advance in deep generative models (DGMs) with a Gaussian distribution as a prior distribution. Ye and Zhao [5] applied VAE to multi-manifold clustering in the scheme of non-parametric Bayesian method and it gave an advantage of realistic image generation in the clustering tasks. Xu et al. [6] adapted VAE to a generative adversarial network [7] in image-to-image translation task to learn many-to-many mapping functions among multiple domains.

If we focus on the VAE, the VAE assumes the prior distribution to be $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ with the learning on the approximated $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$. Also, *Stick-Breaking VAE* (SBVAE) [8] is a nonparametric version of the VAE, which modeled the latent dimension to be infinite with a stick-breaking process [9].

While these VAEs assume that the prior distribution of the latent variables to be continuous random variables, recent studies introduce the approximations on continuous priors for discrete random variables [10]. The key of these approximations is enabling the backpropagation with the reparametrization technique, or the stochastic gradient variational Bayes (SGVB) estimator, while the modeled prior becomes an approximated discrete distribution. The applications of these approximations on discrete random variables' priors include the prior modeling of a multinomial distribution which is frequently used in the probabilistic graphical models (PGMs). Inherently, the multinomial distributions can take a Dirichlet distribution as a conjugate prior, and the demands on such prior have motivated the works like Jang et al. [10], Maddison et al. [11], Rolfe [12] that support the multinomial distribution posterior without explicit modeling on a Dirichlet prior.

When we survey the work with explicit modeling on the Dirichlet prior, we found a frequent approach such as utilizing a softmax Laplace approximation [13]. We argue that this approach has a limitation from the *multi-modality* perspective. The Dirichlet distribution is a multi-modal distribution with parameter set-

* Corresponding author.
E-mail addresses: es345@kaist.ac.kr (W. Joo), wonsung.lee@sk.com (W. Lee), sungrae.park@navercorp.com (S. Park), icmoon@kaist.ac.kr (I.-C. Moon).

https://doi.org/10.1016/j.patcog.2020.107514
0031-3203/

Fig. 1. Illustrated probability simplex with Gaussian-Softmax, GEM, and Dirichlet distributions. Unlike the Gaussian-Softmax or the GEM distribution, the Dirichlet distribution is able to capture the multi-modality that illustrates multiple peaks at the vertices of the probability simplex.

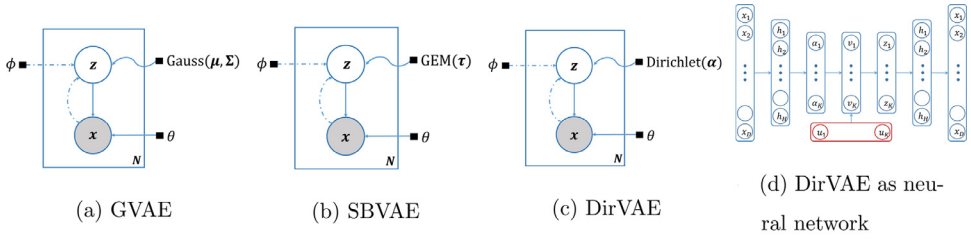


Fig. 2. Sub-Fig. 2a–c are the graphical notations of the VAEs as latent variable models. The solid lines indicate the generative sub-models where the waved lines denote a prior distribution of the latent variables. The dotted lines indicate the inference sub-models. Sub-Fig. 2d denotes a neural network structure corresponding to Sub-Fig. 2c. Red nodes denote the random nodes which allow the backpropagation flows to the input.

tings, see Fig. 1, which is infeasible to generate with the Gaussian distribution with a softmax function. Therefore, the VAEs with a single-modal distribution prior, such as Gaussian VAE, cannot be a perfect substitute for the direct approximation on the Dirichlet distribution.

Utilizing a Dirichlet distribution as a conjugate prior to a multinomial distribution has an advantage compared to the usage of a softmax function on a Gaussian distribution. For instance, Fig. 1 illustrates the potential difficulties in utilizing the softmax function with the Gaussian distribution. Given the three-dimensional probability simplex, the Gaussian-Softmax distribution cannot generate the illustrated case of the Dirichlet distribution with a high probability measure at the vertices of the simplex, i.e. the *multi-modality* where the necessity was emphasized in Hoffman and Johnson [14]. Additionally, the Griffiths-Engen-McCloskey (GEM) distribution [15], which is the prior distribution of SBVAE, is difficult to model the multi-modality because the sampling procedure of the GEM distribution is affected by the *rich-get-richer* phenomenon, so a few components tend to dominate the weight of the samples. This is different from the Dirichlet distribution that does not exhibit such a phenomenon, and the Dirichlet distribution can fairly distribute the weights to the components. Additionally, the Dirichlet distribution is more likely to capture the multi-modality by controlling the prior hyper-parameter, which considered to be important in the latent modeling of texts [1]. Then, we conjecture that an enhanced modeling on the Dirichlet prior is still needed 1) because there are cases that the Gaussian-Softmax approaches, or the softmax Laplace approximation, cannot imitate the Dirichlet distribution; and 2) because the nonparametric approaches could be influenced by the biases that the Dirichlet distribution does not suffer from.

Given these motivations for modeling the Dirichlet distribution with the SGVB estimator, this paper introduces the *Dirichlet Variational Autoencoder* (DirVAE) that shows the same characteristics of the Dirichlet distribution. Naturally, DirVAE is able to model the multi-modal distribution that was not possible with the Gaussian-Softmax and the GEM approaches. These characteristics allow DirVAE to be the prior of the discrete latent distribution, as the original Dirichlet distribution is.

1.2. Contributions

SGVB for Dirichlet Distribution. Introducing DirVAE requires the configuration of the SGVB estimator on the Dirichlet distribution. Specifically, the Dirichlet distribution is a composition of the Gamma random variables, so we approximate the inverse Gamma cumulative distribution function (CDF) with the asymptotic approximation. This approximation on the inverse Gamma CDF becomes the component of approximating the Dirichlet distribution. We compared this approach to the previously suggested approximations, i.e. approaches with the Weibull distribution and with the softmax Gaussian distribution, and our approximation shows the best log-likelihood among the compared approximations.

Resolving Component Collapsing. Moreover, we report that we had to investigate the *component collapsing* along with the research on DirVAE. It has been known that the *component collapsing* issue is resolved by SBVAE because of the meaningful decoder weights from the latent layer to the next layer. However, we found that SBVAE has *latent value collapsing* issue resulting in many near-zero values on the latent dimensions that lead to the incomplete utilization of the latent dimension. Hence, we argue that Gaussian VAE (GVAE) suffers from the *decoder weight collapsing*, previously limitedly defined as *component collapsing*; and SBVAE has a problem of the *latent value collapsing*. Finally, we suggest that the definition of *component collapsing* should be expanded to represent both cases of *decoder weight* and *latent value collapsings*. The proposed DirVAE shows neither the near-zero decoder weights nor the near-zero latent values, so the reconstruction uses the full latent dimension information in most cases. We investigated this issue because our performance gain comes from resolving the expanded version of the *component collapsing*. Due to the component collapsing issues, the existing VAEs have less meaningful latent values or could not effectively use its latent representation. Meanwhile, DirVAE does not have component collapsing due to the multi-modal prior which possibly leads to superior qualitative and quantitative performances. We experimentally showed that DirVAE has more meaningful or disentangled latent representations by image generations and latent value visualizations.

Applications of DirVAE. Technically, the new approximation provides the closed-form loss function derived from the evidence lower bound (ELBO) of DirVAE. The optimization on the ELBO enables the representation learning with DirVAE, and we test the learned representation from DirVAE in two folds. Firstly, we test the representation learning quality by performing the supervised and semi-supervised classification tasks on MNIST, OMNIGLOT, COIL-20, SVHN, and CIFAR-10. These classification tasks conclude that DirVAE has the best classification performances with its learned representation. Secondly, we test the applicability of DirVAE to the existing models, such as topic models with DirVAE priors on 20Newsgroup and RCV1-v2. This experiment shows that the augmentation of DirVAE to the existing neural variational topic models improves the perplexity and the topic coherence, and most of the best performers were DirVAE augmented.

2. Preliminaries

2.1. Variational autoencoders

A VAE is composed of two parts: a generative sub-model and an inference sub-model. In the generative part, a probabilistic decoder reproduces $\hat{\mathbf{x}}$ close to an observation $\mathbf{x}$ from a latent variable $\mathbf{z} \sim p(\mathbf{z})$, i.e. $\mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z}) = p_\theta(\mathbf{x}|\boldsymbol{\zeta})$ where $\boldsymbol{\zeta} = \text{MLP}(\mathbf{z})$ is obtained from a latent variable $\mathbf{z}$ by a multilayer perceptron (MLP). In the inference part, a probabilistic encoder outputs a latent variable $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}) = q_\phi(\mathbf{z}|\boldsymbol{\eta})$ where $\boldsymbol{\eta} = \text{MLP}(\mathbf{x})$ is computed from the observation $\mathbf{x}$ by a MLP. Model parameters, θ and ϕ, are jointly learned by optimizing the below ELBO in Eq. (1) with the stochastic gradient descent method (SGD) through the backpropagations as the ordinary neural networks by using the SGVB estimators on the random nodes.

$$\log p(\mathbf{x}) \geq \mathcal{L}(\mathbf{x}) = \mathbb{E}_{q_{\phi(\mathbf{z}|\mathbf{x})}}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \text{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})) \quad (1)$$

In GVAE [4], the prior distribution of $p(\mathbf{z})$ is assumed to be a standard Gaussian distribution. In SBVAE [8], the prior distribution becomes a GEM distribution that produces samples with a Beta distribution and a stick-breaking algorithm.

2.2. Dirichlet distribution as a composition of Gamma random variables

The Dirichlet distribution is a composition of multiple Gamma random variables. Note that the probability density functions (PDFs) of Dirichlet and Gamma distributions are as follows:

$$\text{Dirichlet}(\mathbf{x};\boldsymbol{\alpha}) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k x_k^{\alpha_k - 1},$$

$$\text{Gamma}(x;\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$$

where $\alpha_k, \alpha, \beta > 0$. In detail, if there are K independent random variables following the Gamma distributions $X_k \sim \text{Gamma}(\alpha_k, \beta)$ or $\mathbf{X} \sim \text{MultiGamma}(\boldsymbol{\alpha}, \beta \cdot \mathbf{1}_K)$ where $\alpha_k, \beta > 0$ for $k = 1, \cdots, K$, then we have $\mathbf{Y} \sim \text{Dirichlet}(\boldsymbol{\alpha})$ where $Y_k = X_k / \sum X_i$. Here, $\mathbf{1}_K$ is all-one K-dimensional vector. It should be noted that the rate parameter, β, should be the same for every Gamma distribution in the composition. Then, the KL divergence is derived as Proposition 1.

Proposition 1. *Define* $\mathbf{X} = (X_1, \cdots, X_K) \sim \text{MultiGamma}(\boldsymbol{\alpha}, \beta \cdot \mathbf{1}_K)$ *as a vector of* K *independent Gamma random variables* $X_k \sim \text{Gamma}(\alpha_k, \beta)$ *where* $\alpha_k, \beta > 0$ *for* $k = 1, \cdots, K$. *The KL divergence between two MultiGamma distributions* $P = \text{MultiGamma}(\boldsymbol{\alpha}, \beta \cdot \mathbf{1}_K)$ *and* $Q = \text{MultiGamma}(\hat{\boldsymbol{\alpha}}, \beta \cdot \mathbf{1}_K)$ *can be derived as the following:*

$$\text{KL}(Q||P) = \sum_k \Big(\log\Gamma(\alpha_k) - \log\Gamma(\hat{\alpha}_k) + (\hat{\alpha}_k - \alpha_k)\psi(\hat{\alpha}_k) \Big) \quad (2)$$

where ψ *is a digamma function.*

Proof. Note that the derivative of a Gamma-like function $\frac{\Gamma(\alpha)}{\beta^\alpha}$ can be derived as follows:

$$\frac{d}{d\alpha}\frac{\Gamma(\alpha)}{\beta^\alpha} = \beta^{-\alpha}(\Gamma'(\alpha) - \Gamma(\alpha)\log\beta) = \int_0^\infty x^{\alpha-1} e^{-\beta x} \log x \, dx.$$

Then, we have the following.

$$\begin{aligned}
\text{KL}(Q||P) &= \int_{\mathcal{D}} q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x})} \, d\mathbf{x} \\
&= \int_0^\infty \cdots \int_0^\infty \prod \text{Gamma}(\hat{\alpha}_k, \beta) \\
&\quad \times \log \frac{\beta^{\sum \hat{\alpha}_k} \prod \Gamma^{-1}(\hat{\alpha}_k) e^{-\beta \sum x_k} \prod x_k^{\hat{\alpha}_k - 1}}{\beta^{\sum \alpha_k} \prod \Gamma^{-1}(\alpha_k) e^{-\beta \sum x_k} \prod x_k^{\alpha_k - 1}} \, d\mathbf{x} \\
&= \int_0^\infty \cdots \int_0^\infty \prod \text{Gamma}(\hat{\alpha}_k, \beta) \\
&\quad \times \Big[\sum (\hat{\alpha}_k - \alpha_k) \log\beta + \sum \log\Gamma(\alpha_k) - \sum \log\Gamma(\hat{\alpha}_k) \\
&\quad + \sum (\hat{\alpha}_k - \alpha_k) \log x_k \Big] \, d\mathbf{x} \\
&= \Big[\sum (\hat{\alpha}_k - \alpha_k) \log\beta + \sum \log\Gamma(\alpha_k) - \sum \log\Gamma(\hat{\alpha}_k) \Big] \\
&\quad + \int_0^\infty \cdots \int_0^\infty \frac{\beta^{\hat{\alpha}_k}}{\prod \Gamma(\hat{\alpha}_k)} e^{-\beta \sum x_k} \\
&\quad \times \prod x_k^{\hat{\alpha}_k - 1} \Big(\sum (\hat{\alpha}_k - \alpha_k) \log x_k \Big) \, d\mathbf{x} \\
&= \Big[\sum (\hat{\alpha}_k - \alpha_k) \log\beta + \sum \log\Gamma(\alpha_k) - \sum \log\Gamma(\hat{\alpha}_k) \Big] \\
&\quad + \sum (\hat{\alpha}_k - \alpha_k) \beta^{\hat{\alpha}_k} \Gamma^{-1}(\hat{\alpha}_k) \beta^{-\hat{\alpha}_k} \Big(\Gamma'(\hat{\alpha}_k) \\
&\quad - \Gamma(\hat{\alpha}_k) \log\beta \Big) \\
&= \sum \log\Gamma(\alpha_k) - \sum \log\Gamma(\hat{\alpha}_k) + \sum (\hat{\alpha}_k - \alpha_k) \psi(\hat{\alpha}_k)
\end{aligned}$$

□

2.3. SGVB for Gamma random variable and approximation on Dirichlet distribution

This section discusses several ways of approximating the Dirichlet random variable; or the SGVB estimators for the Gamma random variables which compose a Dirichlet distribution. Utilizing SGVB requires a differentiable non-centered parametrization (DNCP) for the distribution [16]. The main SGVB for Gamma random variables, used in DirVAE, is using the inverse Gamma CDF approximation explained in the next section. Prior works include two approaches: the use of the Weibull distribution and the softmax Gaussian distribution, and the two approaches are explained in this section.

Approximation with Weibull distribution Because of the similarity between the Weibull distribution and the Gamma distribution PDFs, some prior works used the Weibull distribution as a posterior distribution of the prior Gamma distribution [17]:

$$\text{Weibull}(x;k,\lambda) = \frac{k}{\lambda}\Big(\frac{x}{\lambda}\Big)^{k-1} e^{-(x/\lambda)^k} \text{ where } k, \lambda > 0.$$

The paper Zhang et al. [17] pointed out that there are two useful characteristics when approximating the Gamma distribution with the Weibull distribution. One useful property is that the KL divergence between the Weibull and the Gamma distributions is expressed in a closed form, and the other is the simple reparametrization trick with a closed form of the inverse CDF from the Weibull distribution. However, we noticed that the Weibull distribution has a component of $e^{-(x/\lambda)^k}$, and the Gamma distribution does not have the additional power term of k in the component.

Since k is placed in the exponential component, small changes on k can cause a significant difference that limits the optimization.

Approximation with softmax Gaussian distribution As in Srivastava and Sutton [13], MacKay [18], a Dirichlet distribution can be approximated by a softmax Gaussian distribution by using a softmax Laplace approximation. The relation between the Dirichlet parameter $\boldsymbol{\alpha}$ and the Gaussian parameters $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ is explained as the following:

$$\mu_k = \log \alpha_k - \frac{1}{K}\sum_i \log \alpha_i, \quad \Sigma_k = \frac{1}{\alpha_k}\left(1 - \frac{2}{K}\right) + \frac{1}{K^2}\sum_i \frac{1}{\alpha_i} \tag{3}$$

where $\boldsymbol{\Sigma}$ is assumed to be a diagonal matrix, and we use the reparametrization trick in the usual GVAE for the SGVB estimator.

3. Model description

3.1. Dirichlet Variational Autoencoder (DirVAE)

Generative sub-model The key difference between the generative models between DirVAE and GVAE is the prior distribution assumption on the latent variable $\mathbf{z}$. Instead of using the standard Gaussian distribution, we use the Dirichlet distribution which is a conjugate prior distribution of the multinomial distribution.

$$\mathbf{z} \sim p(\mathbf{z}) = \text{Dirichlet}(\boldsymbol{\alpha}), \quad \mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z}) \tag{4}$$

Inference sub-model The probabilistic encoder with an approximating posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$ is designed to be Dirichlet($\hat{\boldsymbol{\alpha}}$). The approximated posterior parameter $\hat{\boldsymbol{\alpha}}$ is derived by the MLP from the observation $\mathbf{x}$ in dataset $\mathcal{D}$ with positive output function such as softplus function, so the outputs can be positive values constrained by the Dirichlet distribution. Here, we do not directly sample $\mathbf{z}$ from the Dirichlet distribution. Instead, we use the Gamma composition method described in Section 2.2. Firstly, we draw $\mathbf{v} \sim \text{MultiGamma}(\boldsymbol{\alpha}, \beta \cdot \mathbf{1}_K)$. Then, we normalize $\mathbf{v}$ with its summation $\sum v_i$.

Objective function The objective function to optimize the model parameters, θ and ϕ, is composed of Eqs. (1) and (2). Eq. (5) is the loss function to optimize after the composition. The inverse Gamma CDF method explained in the next paragraph enables the backpropagation flows to the input with SGD. Here, for the fair comparison of expressing the Dirichlet distribution between the inverse Gamma CDF approximation method and the softmax Gaussian method, we set $\alpha_k = 1 - 1/K$ when $\mu_k = 0$ and $\Sigma_k = 1$ by using Eq. (3); and $\beta = 1$.

$$\mathcal{L}(\mathbf{x}) = \mathbb{E}_{q_{\phi(\mathbf{z}|\mathbf{x})}}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \sum_k \Big(\log\Gamma(\alpha_k) - \log\Gamma(\hat{\alpha}_k) + (\hat{\alpha}_k - \alpha_k)\psi(\hat{\alpha}_k)\Big) \tag{5}$$

Approximation with inverse Gamma CDF A previous work Knowles [19] suggested that, if $X \sim \text{Gamma}(\alpha, \beta)$, and if $F(x; \alpha, \beta)$ is a CDF of the random variable X, the inverse CDF can be approximated as $F^{-1}(u; \alpha, \beta) \approx \beta^{-1}(u\alpha\Gamma(\alpha))^{1/\alpha}$. Hence, we can introduce an auxiliary variable $u \sim \text{Uniform}(0, 1)$ to take over all the randomness of X, and we treat the Gamma sampled X as a deterministic value in terms of α and β. The whole learning procedure is also summarized in Table 1.

It should be noted that there has been a practice of utilizing the combination of decomposing a Dirichlet distribution and approximating each Gamma component with inverse Gamma CDF. However, such practices have not been examined with their learning properties and applicabilities. The following sections show a new aspect of *component collapsing* that can be remedied by this combination on Dirichlet prior in VAE, and the sections illustrate the performance gains in a certain set of applications, i.e. topic modeling.

Table 1
Parameter update process of DirVAE.

```
Initialize neural network parameter φ.
repeat
    for x in dataset D
        Encode x into α̂ = Encoder(x).
        Sample u ~ Uniform(0, 1)^K.
        Obtain v = Approx. CDF^{-1}(u).
        Normalize v into sum-to-one z.
        Decode z into x̂ = Decoder(z).
        Compute ELBO L(x).
        Update neural network parameter φ via SGD.
        if hyper-parameter update is True
            Update hyper-parameter α via MME.
until convergence.
```

3.2. Hyper-parameter learning strategy of DirVAE

In this section, we introduce the method of moment estimator (MME) to update the Dirichlet prior parameter $\boldsymbol{\alpha}$. The detailed update equation can be found in Proposition 2. After the burn-in period for stabilizing the neural network parameters, we use the MME for the hyper-parameter learning using the sampled latent values during training. We alternatively update the neural network parameters and hyper-parameter $\boldsymbol{\alpha}$. Once we update hyper-parameter $\boldsymbol{\alpha}$ for some periods, by fixing the learned hyper-parameter, finalize the learning process with additional neural network parameter updates. We choose this estimator because of its closed form nature and consistency [20].

Proposition 2. *Given a proportion set* $\mathcal{D} = \{\mathbf{p}_1, \cdots, \mathbf{p}_N\}$ *sampled from* Dirichlet($\boldsymbol{\alpha}$), *MME of the hyper-parameter* $\boldsymbol{\alpha}$ *is as the following:*

$$\alpha_k \leftarrow \frac{S}{N}\sum_n p_{n,k} \text{ where } S = \frac{1}{K}\sum_k \frac{\tilde{\mu}_{1,k} - \tilde{\mu}_{2,k}}{\tilde{\mu}_{2,k} - \tilde{\mu}_{1,k}^2}$$

$$\text{for } \tilde{\mu}_{j,k} = \frac{1}{N}\sum_n p_{n,k}^j.$$

Proof. Define $\mu_{j,k} = \mathbb{E}[p_k^j]$ as the jth moment of the kth dimension of Dirichlet distribution with prior $\boldsymbol{\alpha}$. Then, by the law of large number, $\mu_{j,k} \approx \tilde{\mu}_{j,k}$. It is well-known that $\mu_{1,k} = \frac{\alpha_k}{\sum_i \alpha_i}$ and $\mu_{2,k} = \frac{\alpha_k}{\sum_i \alpha_i}\frac{1+\alpha_k}{1+\sum_i \alpha_i} = \mu_{1,k}\frac{1+\alpha_k}{1+\sum_i \alpha_i}$ so that

$$\text{numerator}\left(\frac{\mu_{1,k} - \mu_{2,k}}{\mu_{2,k} - \mu_{1,k}^2}\right) = \frac{\alpha_k}{\sum_i \alpha_i} - \frac{\alpha_k}{\sum_i \alpha_i}\frac{1+\alpha_k}{1+\sum_i \alpha_i} = \frac{\alpha_k(\sum_{i\neq k}\alpha_i)}{(\sum_i \alpha_i)(1+\sum_i \alpha_i)}$$

$$\text{denominator}\left(\frac{\mu_{1,k} - \mu_{2,k}}{\mu_{2,k} - \mu_{1,k}^2}\right) = \frac{\alpha_k}{\sum_i \alpha_i}\frac{1+\alpha_k}{1+\sum_i \alpha_i} - \left(\frac{\alpha_k}{\sum_i \alpha_i}\right)^2 = \frac{\alpha_k(\sum_{i\neq k}\alpha_i)}{(\sum_i \alpha_i)^2(1+\sum_i \alpha_i)}$$

holds for each $k = 1, \cdots, K$. Therefore,

$$\sum_i \alpha_i = \frac{\mu_{1,k} - \mu_{2,k}}{\mu_{2,k} - \mu_{1,k}^2} \approx \frac{1}{K}\sum_k \frac{\mu_{1,k} - \mu_{2,k}}{\mu_{2,k} - \mu_{1,k}^2} \approx \frac{1}{K}\sum_k \frac{\tilde{\mu}_{1,k} - \tilde{\mu}_{2,k}}{\tilde{\mu}_{2,k} - \tilde{\mu}_{1,k}^2}$$

and hence,

$$\hat{\alpha}_k = \left(\sum_i \alpha_i\right)\tilde{\mu}_{1,k} = \frac{S}{N}\sum_n p_{n,k}.$$

□

4. Baseline models and datasets

4.1. Baseline models

We select the following models as baselines of DirVAE: 1) the standard GVAE; 2) GVAE with softmax (GVAE-Softmax) approximating the Dirichlet distribution with the softmax Gaussian distribution; 3) SBVAE with the Kumaraswamy distribution (SBVAE-Kuma) & the Gamma composition (SBVAE-Gamma) described in Nalisnick and Smyth [8]; and 4) DirVAE with the Weibull distribution (DirVAE-Weibull) approximating the Gamma distribution with the Weibull distribution described in Zhang et al. [17].

4.2. Datasets

We use the following benchmark datasets for the experiments: 1) MNIST; 2) MNIST with rotations (MNIST + rot); 3) OMNIGLOT; 4) COIL-20; 5) SVHN with PCA transformation; and 6) CIFAR-10 with PCA transformation. MNIST [21] is a hand-written digit grayscale image dataset of size 28 $\times$ 28 with 10 labels, consists of 60,000 training data and 10,000 testing data. MNIST + rot data is reproduced by the authors of Nalisnick and Smyth [8] consists of MNIST and rotated MNIST. OMNIGLOT [22] is another hand-written grayscale image dataset of characters with 28 $\times$ 28 size and 50 labels which can be considered as more complex dataset than the MNISTs, consists of 24,345 training data and 8,070 testing data. COIL-20 [23] is a grayscale Columbia University Image Library dataset of 20 different objects in 72 viewpoints by rotating the viewpoints for each object, however, it has 128 $\times$ 128 size which is larger than MNISTs and OMNIGLOT. We randomly divide the dataset into 48 training data and 24 testing data for each object. SVHN [24] is a Street View House Numbers image dataset with the dimension-reduction by PCA into 500 dimensions [8]. CIFAR-10 [25] is a Canadian Institute For Advanced Research image dataset with the dimension-reduction by PCA into 891 dimensions following the similar pre-processing procedure in Nalisnick and Smyth [8].

5. Experiments and performance evaluations as a pure VAE

This section reports the experimental results with the following experiment settings: 1) a pure VAE model; and 2) a component collapsing issue.

5.1. Experimental setting

As a pure VAE model, we compare DirVAE with the following models: GVAE, GVAE-Softmax, SBVAE-Kuma, SBVAE-Gamma, and DirVAE-Weibull. We divided the datasets into {train,valid,test} as the following: MNIST $= \{45,000 : 5,000 : 10,000\}$ and OMNIGLOT $= \{22,095 : 2,250 : 8,070\}$. For MNIST, we use 50-dimension latent variables with two hidden layers in the encoder and one hidden layer in the decoder of 500 dimensions. We set $\boldsymbol{\alpha} = 0.98 \cdot \mathbf{1}_{50}$ for the fair comparison to GVAEs using the Eq. (3). The batch size was set to be 100. For OMNIGLOT, we use 100-dimension latent variables with two hidden layers in the encoder and one hidden layer in the decoder of 500 dimensions. We assume $\boldsymbol{\alpha} = 0.99 \cdot \mathbf{1}_{100}$ for the fair comparison to the GVAEs using the Eq. (3). The batch size was set to be 15.

For both datasets, the gradient clipping is used; ReLU function [26] is used as an activation function in hidden layers; Xavier initialization [27] is used for the neural network parameter initialization; and the Adam optimizer [28] is used as an optimizer with learning rate `5e-4` for all VAEs except `3e-4` for the SBVAEs. The prior assumptions for each VAE is the following: 1) $\mathcal{N}(\mathbf{0}, \mathbf{I})$ for GVAE and GVAE-Softmax; 2) GEM(5) for the SBVAEs; and 3) Dirichlet($0.98 \cdot \mathbf{1}_{50}$) (MNIST) and Dirichlet($0.99 \cdot \mathbf{1}_{100}$) (OMNIGLOT) for DirVAE-Weibull. Finally, to compute the marginal log-likelihood, we used 100 samples for each 1,000 randomly selected from the test data.

Additionally, DirVAE-Learning uses the same hyper-parameter $\boldsymbol{\alpha}$ for the initial value, but DirVAE-Learning optimizes hyper-parameter $\boldsymbol{\alpha}$ by the following stages through the learning iterations using the MME method in Section 3.2: 1) the burn-in period for stabilizing the neural network parameters; 2) the alternative update period for the neural network parameters and hyper-parameter $\boldsymbol{\alpha}$; and 3) the update period for the neural network parameters with the fixed learned hyper-parameter $\boldsymbol{\alpha}$.

5.2. Quantitative result

For the quantitative comparison among the VAEs, we calculated the Monte-Carlo estimation on the marginal negative log-likelihood, the negative ELBO, and the reconstruction loss. The marginal log-likelihood is approximated as $p(\mathbf{x}) \approx \sum_i \frac{p(\mathbf{x}|\mathbf{z}_i)p(\mathbf{z}_i)}{q(\mathbf{z}_i)}$ for a single instance $\mathbf{x}$ where $q(\mathbf{z})$ is a posterior distribution of a prior distribution $p(\mathbf{z})$, which is further derived in Proposition 3.

Proposition 3. *For a single instance* $\mathbf{x}$ *in dataset* $\mathcal{D}$*, the marginal log-likelihood is approximated as* $p(\mathbf{x}) \approx \sum_i \frac{p(\mathbf{x}|\mathbf{z}_i)p(\mathbf{z}_i)}{q(\mathbf{z}_i)}$*, where* $q(\mathbf{z})$ *is a posterior distribution of a prior distribution* $p(\mathbf{z})$*.*

Proof.

$$
\begin{aligned}
p(\mathbf{x}) &= \int_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \int_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) \frac{q(\mathbf{z})}{q(\mathbf{z})} d\mathbf{z} \\
&= \int_{\mathbf{z}} p(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) \frac{q(\mathbf{z})}{q(\mathbf{z})} d\mathbf{z} = \int_{\mathbf{z}} \frac{p(\mathbf{x}|\mathbf{z}) p(\mathbf{z})}{q(\mathbf{z})} q(\mathbf{z}) d\mathbf{z} \\
&\approx \sum_i \frac{p(\mathbf{x}|\mathbf{z}_i) p(\mathbf{z}_i)}{q(\mathbf{z}_i)} \text{ where } \mathbf{z}_i \sim q(\mathbf{z})
\end{aligned}
$$

□

Table 2 shows the overall performance of VAEs. DirVAE outperforms all baselines in both datasets from the log-likelihood perspective. Also, the experiment in OMNIGLOT dataset which is more challenging than MNIST, DirVAE outperforms all baselines in all performance measures. The advantage of DirVAE comes from the better encoding of the latent variables that can be used for classification tasks which we examine in the next section. While DirVAE-Weibull follows the prior modeling with the Dirichlet distribution, the Weibull based approximation can be improved by adopting the proposed approach with the inverse Gamma CDF. Also, by comparing DirVAE and DirVAE-Learning, we experimentally show that the hyper-parameter learning introduced in Section 3.2 improves the performance in all measures. Finally, we present the learned hyper-parameter $\boldsymbol{\alpha}$ in Fig. 3.

5.3. Qualitative result

As a qualitative result, we report the latent dimension-wise reconstructions which are decoder outputs with each one-hot vector in the latent dimension. Fig. 4a shows 50 reconstructed images corresponding to each latent dimension from GVAE-Softmax, SBVAE, and DirVAE. We manually ordered the digit-like figures in the ascending order for GVAE-Softmax and DirVAE. We can see that GVAE-Softmax and SBVAE have components without significant semantic information, which we will discuss further in Section 5.4, and DirVAE has interpretable latent dimensions in most of the latent dimensions. Fig. 4b also supports the quality of the disentangled latent representations from DirVAE by visualizing the whole learned latent values through t-SNE [29].

Table 2
Negative log-likelihood, negative ELBO, and reconstruction loss of the VAEs for MNIST and OMNIGLOT dataset. The lower values are the better for all measures.

	MNIST ($K = 50$)			OMNIGLOT ($K = 100$)		
	Neg. LL	Neg. ELBO	Reconst. Loss	Neg. LL	Neg. ELBO	Reconst. Loss
GVAE	$94.54_{\pm 0.79}$	$\mathbf{98.58_{\pm 0.04}}$	$\mathbf{74.31_{\pm 0.13}}$	$119.29_{\pm 0.44}$	$126.42_{\pm 0.24}$	$98.90_{\pm 0.36}$
GVAE-Softmax	$98.18_{\pm 0.61}$	$103.49_{\pm 0.16}$	$79.36_{\pm 0.82}$	$130.01_{\pm 1.16}$	$139.73_{\pm 0.81}$	$123.34_{\pm 1.43}$
SBVAE-Kuma	$99.27_{\pm 0.48}$	$102.60_{\pm 1.81}$	$83.90_{\pm 0.82}$	$130.73_{\pm 2.17}$	$132.86_{\pm 3.03}$	$119.25_{\pm 1.00}$
SBVAE-Gamma	$102.14_{\pm 0.69}$	$135.30_{\pm 0.24}$	$113.89_{\pm 0.25}$	$128.82_{\pm 1.82}$	$149.30_{\pm 0.82}$	$136.36_{\pm 1.53}$
DirVAE-Weibull	$114.59_{\pm 11.15}$	$183.33_{\pm 2.96}$	$150.92_{\pm 3.70}$	$140.89_{\pm 3.21}$	$198.01_{\pm 2.46}$	$145.52_{\pm 3.13}$
DirVAE	$\mathbf{87.64_{\pm 0.64}}$	$100.47_{\pm 0.35}$	$81.50_{\pm 0.27}$	$\mathbf{104.42_{\pm 0.73}}$	$\mathbf{118.84_{\pm 0.02}}$	$\mathbf{96.75_{\pm 0.24}}$
DirVAE-Learning	$\mathbf{84.76_{\pm 0.26}}$	$\mathbf{98.32_{\pm 0.18}}$	$79.56_{\pm 0.09}$	$\mathbf{95.82_{\pm 0.26}}$	$\mathbf{116.33_{\pm 0.25}}$	$\mathbf{95.11_{\pm 0.41}}$

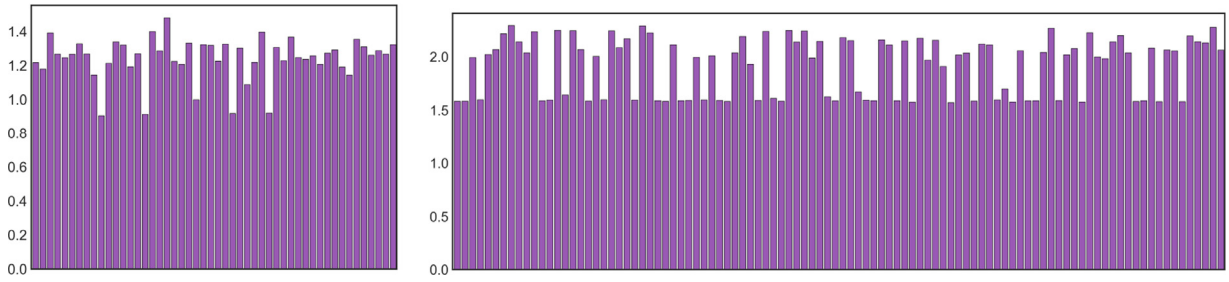


Fig. 3. The optimized dimension-wise $\boldsymbol{\alpha}$ values from DirVAE-Learning with (Left) MNIST, and (Right) OMNIGLOT.

(a) Latent dimension-wise reconstructions of GVAE-Softmax, SBVAE, and DirVAE. DirVAE shows more meaningful latent dimensions than other VAEs.

(b) t-SNE latent embeddings of (Left) GVAE, (Middle) SBVAE, (Right) DirVAE.

Fig. 4. Latent dimension visualization with reconstruction images and t-SNE.

5.4. Discussion on component collapsing

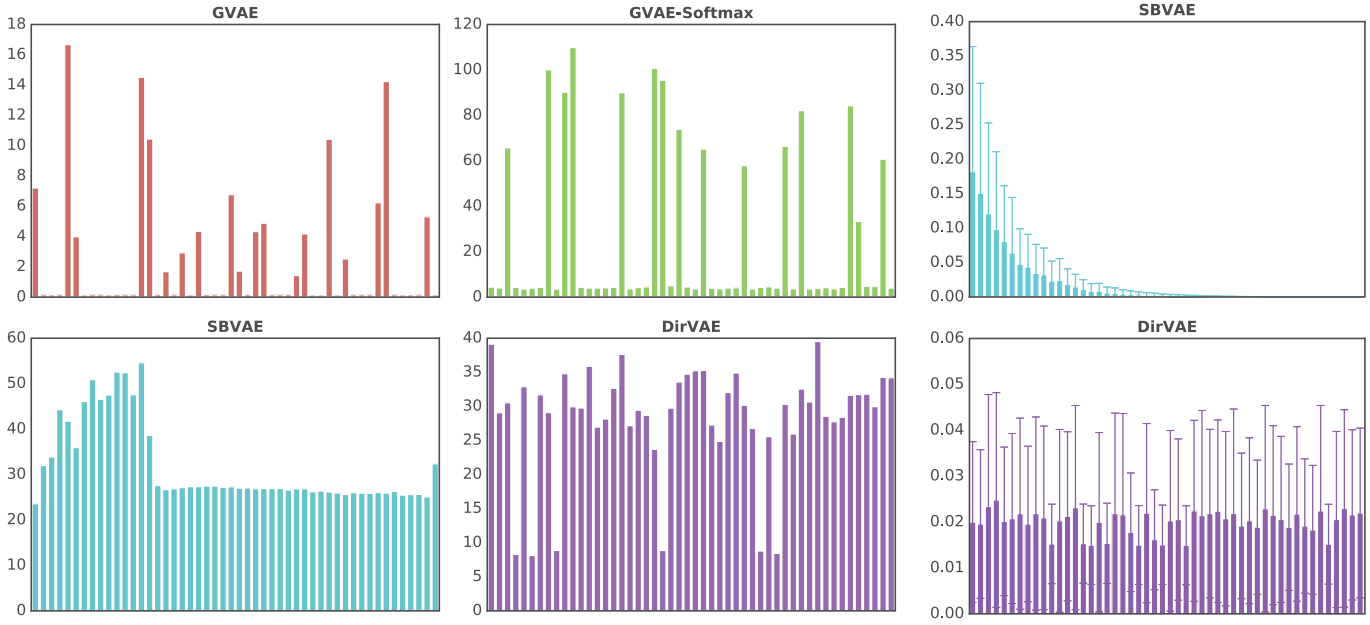
5.4.1. Decoder weight collapsing, a.k.a. component collapsing

One main issue of GVAE is *component collapsing* that there are a significant number of near-zero decoder weights from the latent neurons to the next decoder neurons. If these weights become near-zero, the values of the latent dimensions loose influence to the next decoder, and this means inefficient learning given a neural network structure. The same issue occurs when we use GVAE-Softmax. We rename this component collapsing phenomenon as *decoder weight collapsing* to specifically address the collapsing source.

5.4.2. Latent value collapsing

SBVAE claims that SBVAE solved the *decoder weight collapsing* by learning the meaningful weights as shown in Fig. 5a. However, we notice that SBVAE produces the output values, not the weight parameters, from the latent dimension to be near-zero in many latent dimensions after averaging many samples obtained from the test dataset. Fig. 5b shows the properties of DirVAE and SBVAE from the perspective of the latent value collapsing, which SBVAE shows many near-zero average means and near-zero average variances, while DirVAE does not. The average Fisher kurtosis and average skewness of DirVAE are 5.76 and 2.03, respectively over the dataset, while SBVAE has 20.85 and 4.35, which states that the latent output distribution from SBVAE is more skewed than that of DirVAE. We found out that these near-zero latent values prevent learning on decoder weights, which we introduce as another type of collapsing problem, as *latent value collapsing* that is different from the *decoder weight collapsing*. These results mean that SBVAE distributes a few meaningful, or non-near-zero, latent values sparsely over a few dimensions while DirVAE samples relatively dense latent values. In other words, DirVAE utilizes the full spectrum of latent dimensions compared to SBVAE, and DirVAE has a better learning capability in the decoder network. Fig. 4a supports the argument on the latent value collapsing by activating each and single latent dimension with a one-hot vector through the decoder. The non-changing latent dimension-wise images of SBVAE proves that there were no generation differences between the two differently activated one-hot latent values.

The paper proposing *Importance Weighted Autoencoder* (IWAE) [30] also pointed out that GVAE does not fully activate its latent dimensions and wastes its capacity of the neural network structure. One way of counting the number of active dimensions of VAEs suggested by the authors of Burda et al. [30] is using the latent dimension-wise variance across the dataset by assuming that a single latent dimension is activated if $\mathrm{Cov}_{\mathbf{x}}(\mathbb{E}_{q(z_k|\mathbf{x})}[z_k])$ is greather than a threshold $\delta > 0$ where the authors of Burda



(a) Latent dimension-wise L2-norm of decoder weights of VAEs.

(b) Average latent values of VAEs.

Fig. 5. Fig. 5 a shows GVAE and GVAE-Softmax have component collapsing issue, while SBVAE and DirVAE do not. Fig. 5b shows that SBVAE has many near-zero output values in the latent dimensions.

Table 3
The number of relative active units in latent dimension of VAEs.

	MNIST ($K = 50$, $\delta = 1e{-}4$)	OMNIGLOT ($K = 100$, $\delta = 2.5e{-}5$)	COIL-20 ($K = 50$, $\delta = 1e{-}4$)
GVAE	18.0 ± 0.6	29.8 ± 0.7	24.8 ± 1.6
GVAE-Softmax	16.8 ± 1.2	13.2 ± 1.3	26.2 ± 2.9
SBVAE	15.4 ± 0.5	17.8 ± 0.4	28.8 ± 3.5
DirVAE	**45.2 ± 0.4**	**63.6 ± 1.5**	**47.4 ± 1.5**

et al. [30] set $\delta = 1e{-}2$. However, this measure cannot be directly applied to other VAEs which have additional *sum-to-one* constraint in the latent variables such as GVAE-Softmax, SBVAE, and DirVAE. Since variance is quadratically proportioned to the scale, such sum-to-one constraint will downscale the variance of latent variables, while GVAE does not have any upper or lower bounds on its latent values. Hence, we rather focus on *relative activeness* across the latent dimensions by suggesting an alternative measurement for GVAE which is taking absolute value and then normalizing with its summation, i.e. from $\mathbf{z} = (z_1, \cdots, z_K)$ to $\hat{\mathbf{z}} = (|z_1| \big/ \sum_k |z_k|, \cdots, |z_K| \big/ \sum_k |z_k|)$. Once we transform the latent values of GVAE in sum-to-one form to see the relative activeness, we find a proper threshold δ that gives a similar number of active dimensions with the original measurement.

Table 3 enumerates the number of relative active units of VAEs with the chosen δ values for each dataset. One can see that the number of relative active units is similar to the number of meaningful latent dimensions, which is being observed by the latent dimension-wise reconstructions in Fig. 4a. This implies that the better qualitative results of DirVAE are enabled by the wider utilization on the latent dimensions than any other VAEs.

6. Applications of DirVAE

This section reports the experimental results with the following experiment settings: 1) a supervised classification task with VAEs; 2) a semi-supervised classification task with VAEs; and 3) topic models with DirVAE augmentations.

6.1. Application 1: Experiments of supervised classification with VAEs

6.1.1. Experiment settings

We tested the performance of the supervised classification task with the learned latent representation from the VAEs. We applied the vanilla version of VAEs to the datasets, and we classified the latent representation of instances with k-Nearest Neighbor (kNN) which is one of the simplest classification algorithms. Hence, this experiment can better distinguish the performance of the disentangled representation learning in the classification task. For the supervised classification task on the latent representation of the VAEs, we used exactly the same experimental settings as in Section 5.1.

6.1.2. Experiment results

Table 4 enumerates the performances from the experimented VAEs in the datasets of MNIST, OMNIGLOT, and COIL-20. All datasets indicated that DirVAE shows the best performance in reducing the classification error, which we conjecture that the performance is gathered from the better representation learning. We identified that the classification with OMNIGLOT is difficult given that the kNN error rates with the raw original data are as high as 69.94%, 69.41%, and 70.10%. This high error rate mainly origi-

Table 4
The error rate of kNN with the latent representations of VAEs.

	MNIST ($K = 50$)			OMNIGLOT ($K = 100$)			COIL-20 ($K = 50$)		
	$k = 3$	$k = 5$	$k = 10$	$k = 3$	$k = 5$	$k = 10$	$k = 3$	$k = 5$	$k = 10$
GVAE	$27.16_{\pm 0.48}$	$20.20_{\pm 0.93}$	$14.89_{\pm 0.40}$	$92.34_{\pm 0.25}$	$91.21_{\pm 0.18}$	$88.79_{\pm 0.35}$	$5.75_{\pm 0.76}$	$6.92_{\pm 1.06}$	$8.79_{\pm 0.73}$
GVAE-Softmax	$25.68_{\pm 2.64}$	$21.79_{\pm 2.17}$	$18.75_{\pm 2.06}$	$94.76_{\pm 0.20}$	$94.22_{\pm 0.37}$	$92.98_{\pm 0.42}$	$14.62_{\pm 0.77}$	$15.79_{\pm 0.49}$	$17.58_{\pm 0.80}$
SBVAE	$10.01_{\pm 0.52}$	$9.58_{\pm 0.47}$	$9.39_{\pm 0.54}$	$86.90_{\pm 0.82}$	$85.10_{\pm 0.89}$	$82.96_{\pm 0.64}$	$17.33_{\pm 1.20}$	$18.12_{\pm 0.32}$	$18.92_{\pm 0.40}$
DirVAE	$\mathbf{5.98_{\pm 0.06}}$	$\mathbf{5.29_{\pm 0.06}}$	$\mathbf{5.06_{\pm 0.06}}$	$\mathbf{76.55_{\pm 0.23}}$	$\mathbf{73.81_{\pm 0.29}}$	$\mathbf{70.95_{\pm 0.29}}$	$\mathbf{5.17_{\pm 0.77}}$	$\mathbf{6.37_{\pm 0.43}}$	$\mathbf{8.42_{\pm 0.43}}$

Table 5
The error rate of semi-supervised classification task using VAEs.

	MNIST ($K = 50$)			SVHN ($K = 50$)			
	10%	5%	1%	25%	10%	5%	1%
GVAE [8]	$\mathbf{3.95_{\pm 0.15}}$	$\mathbf{4.74_{\pm 0.43}}$	$11.55_{\pm 2.28}$	$34.50_{\pm 5.74}$	$36.08_{\pm 1.49}$	$48.75_{\pm 1.47}$	$69.58_{\pm 1.64}$
SBVAE [8]	$4.86_{\pm 0.14}$	$5.29_{\pm 0.39}$	$7.34_{\pm 0.47}$	$30.35_{\pm 0.96}$	$32.08_{\pm 4.00}$	$37.07_{\pm 5.22}$	$61.37_{\pm 3.60}$
DirVAE	$4.60_{\pm 0.07}$	$5.05_{\pm 0.18}$	$\mathbf{7.00_{\pm 0.17}}$	$\mathbf{22.10_{\pm 1.03}}$	$\mathbf{24.81_{\pm 1.13}}$	$\mathbf{28.45_{\pm 1.14}}$	$\mathbf{55.99_{\pm 3.30}}$
	MNIST + rot ($K = 50$)			CIFAR-10 ($K = 50$)			
	10%	5%	1%	25%	10%	5%	1%
GVAE [8]	$21.78_{\pm 0.73}$	$27.72_{\pm 0.69}$	$38.13_{\pm 0.95}$	$62.51_{\pm 0.65}$	$65.97_{\pm 0.86}$	$70.48_{\pm 3.20}$	$80.49_{\pm 0.95}$
SBVAE [8]	$11.78_{\pm 0.39}$	$14.27_{\pm 0.58}$	$27.67_{\pm 1.39}$	$64.07_{\pm 1.24}$	$67.28_{\pm 0.30}$	$71.69_{\pm 2.92}$	$77.70_{\pm 4.95}$
DirVAE	$\mathbf{11.18_{\pm 0.32}}$	$\mathbf{13.53_{\pm 0.46}}$	$\mathbf{26.20_{\pm 0.66}}$	$\mathbf{57.86_{\pm 0.32}}$	$\mathbf{62.11_{\pm 1.65}}$	$\mathbf{64.52_{\pm 1.24}}$	$\mathbf{70.72_{\pm 0.76}}$

Table 6
Hyper-parameter selections for DirVAE augmentations.

	20Newsgroups ($K = 50$)			RCV1-v2 ($K = 100$)		
	ProdLDA	NVDM	GSM	ProdLDA	NVDM	GSM
Add DirVAE	$0.98 \cdot \mathbf{1}_{50}$	$0.95 \cdot \mathbf{1}_{50}$	$0.20 \cdot \mathbf{1}_{50}$	$0.99 \cdot \mathbf{1}_{100}$	$0.90 \cdot \mathbf{1}_{100}$	$0.01 \cdot \mathbf{1}_{100}$

Table 7
Topic modeling performances of perplexity and NPMI with DirVAE augmentations.

		20Newsgroups ($K = 50$)				RCV1-v2 ($K = 100$)			
		ProdLDA	NVDM	GSM	LDA (Gibbs)	ProdLDA	NVDM	GSM	LDA (Gibbs)
Perplexity	Reproduced	$1219_{\pm 8.87}$	$810_{\pm 2.60}$	$954_{\pm 1.22}$	$1314_{\pm 18.50}$	$1190_{\pm 45.24}$	$\mathbf{796_{\pm 6.24}}$	$1386_{\pm 21.06}$	$1126_{\pm 12.66}$
	Add SBVAE	$1164_{\pm 2.55}$	$878_{\pm 14.21}$	$980_{\pm 13.50}$	-	$1077_{\pm 22.57}$	$1050_{\pm 12.19}$	$1670_{\pm 4.78}$	-
	Add DirVAE	$1114_{\pm 2.30}$	$\mathbf{752_{\pm 12.17}}$	$916_{\pm 1.64}$	-	$992_{\pm 2.19}$	$809_{\pm 12.60}$	$1526_{\pm 6.11}$	-
NPMI	Reproduced	$0.273_{\pm 0.019}$	$0.119_{\pm 0.003}$	$0.199_{\pm 0.006}$	$0.225_{\pm 0.002}$	$0.194_{\pm 0.005}$	$0.023_{\pm 0.002}$	$0.267_{\pm 0.019}$	$0.266_{\pm 0.006}$
	Add SBVAE	$0.247_{\pm 0.015}$	$0.162_{\pm 0.007}$	$0.162_{\pm 0.006}$	-	$0.190_{\pm 0.006}$	$0.116_{\pm 0.016}$	$0.207_{\pm 0.004}$	-
	Add DirVAE	$\mathbf{0.359_{\pm 0.026}}$	$0.247_{\pm 0.010}$	$0.201_{\pm 0.003}$	-	$0.193_{\pm 0.004}$	$0.131_{\pm 0.015}$	$\mathbf{0.308_{\pm 0.005}}$	-

nates from the number of classification categories which is 50 categories in our test setting of OMNIGLOT, compared to 10 categories in MNIST. Meanwhile, the low error rate of COIL-20 is due to the composition of COIL-20 dataset which is composed of the same objects with slight rotations.

6.2. Application 2: Experiments of semi-supervised classification with VAEs

6.2.1. Experiment settings

There is a previous work demonstrating that SBVAE outperforms GVAE in semi-supervised classification task [8]. The overall model structure for this semi-supervised classification task uses a VAE with separate random variables, a latent variable $\mathbf{z}$ and a class label variable $\mathbf{y}$, which is introduced as the *M2* model in the original VAE work [31]. However, the same task with SBVAE uses a different model modified to ignore the relation between the class label variable $\mathbf{y}$ and the latent variable $\mathbf{z}$, but they still share the same parent nodes: $q_\phi(\mathbf{z}, \mathbf{y}|\mathbf{x}) = q_\phi(\mathbf{z}|\mathbf{x})q_\phi(\mathbf{y}|\mathbf{x})$ where $q_\phi(\mathbf{y}|\mathbf{x})$ is a discrimitive network for the unseen labels. Finally, below are the objective functions to optimize for the labeled and the unlabeled instances of the semi-supervised classification task, respectively:

$$\begin{aligned}\log p(\mathbf{x}, \mathbf{y}) &\geq \mathcal{L}_{\text{labeled}}(\mathbf{x}, \mathbf{y}) \\ &= \mathbb{E}_{q_{\phi(\mathbf{z}|\mathbf{x})}}[\log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{y})] - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})) \\ &\quad + \log q_\phi(\mathbf{y}|\mathbf{x}), \end{aligned} \tag{6}$$

$$\begin{aligned}\log p(\mathbf{x}) &\geq \mathcal{L}_{\text{unlabeled}}(\mathbf{x}) \\ &= \mathbb{E}_{q_{\phi(\mathbf{z},\mathbf{y}|\mathbf{x})}}[\log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{y}) + \mathbb{H}(q_\phi(\mathbf{y}|\mathbf{x}))] \\ &\quad - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})). \end{aligned} \tag{7}$$

In the above, $\mathbb{H}$ is an entropy function. The actual training on the semi-supervised learning optimizes the weighted sum of Eqs. (6) and (7) with a ratio hyper-parameter $0 < \lambda < 1$.

The datasets are divided into {train, valid, test} as the following: MNIST $= \{45,000 : 5,000 : 10,000\}$, MNIST + rot $= \{70,000 : 10,000 : 20,000\}$, SVHN $= \{65,000 : 8,257 : 26,032\}$, and CIFAR-10 $= \{45,000 : 5,000 : 10,000\}$. For SVHN, dimension reduction into 500 dimensions by PCA is applied as pre-processing. For CIFAR-10, dimension reduction into 891 dimensions by PCA is applied as pre-processing.

Fundamentally, we applied the same experimental settings to GVAE, SBVAE, and DirVAE in this experiment, as specified by the authors in Nalisnick and Smyth [8]. Specifically, the three VAEs

Fig. 6. 20Newsgroups latent document embedding visulaization with t-SNE of topic model augmentations with DirVAE, (Left) ProdLDA, (Middle) NVDM, (Right) GSM.

Table 8
Sample of learned per topic top-10 high probability words from 20Newsgroups with DirVAE augmentation by activating single latent dimensions.

ProdLDA + DirVAE					NVDM + DirVAE					GSM + DirVAE				
season	ide	spacecraft	knife	enforcement	armenian	jesus	hitter	seat	patient	israel	price	internet	team	image
defensive	scsi	satellite	handgun	privacy	azerbaijan	bible	season	gear	disease	israeli	sell	mail	play	ftp
puck	scsus	solar	assault	encrypt	armenia	christianity	braves	rear	doctor	jews	new	computer	game	file
playoff	controller	shuttle	gun	rsa	genocide	doctrine	pitcher	tire	treatment	attack	sale	send	hockey	version
coach	motherboard	nasa	batf	ripem	armenians	scripture	baseball	honda	symptom	world	offer	list	nhl	server
score	isa	mission	criminal	wiretap	turkish	eternal	pitch	oil	medical	jewish	pay	fax	score	program
flyers	cache	professor	homicide	encryption	militia	belief	game	front	health	article	buy	phone	first	system
nhl	mb	lunar	firearm	cipher	massacre	christian	player	mile	hospital	arab	good	email	division	software
team	floppy	orbit	police	cryptography	village	faith	defensive	wheel	pain	peace	condition	address	go	package
ice	ram	rocket	apartment	escrow	turks	resurrection	team	engine	medicine	land	money	information	win	support

used the same network structures of 1) a hidden layer of 500 dimension for MNIST; and 2) four hidden layers of 500 dimensions for MNIST + rot, SVHN and CIFAR-10 with the residual network for the last three hidden layers. The latent variables have 50 dimensions for all settings. The ratio parameter λ is set to be 0.375 for the MNISTs, and 0.45 for SVHN and CIFAR-10. ReLU function is used as an activation function in hidden layers, and the neural network parameters were initialized by sampling from $\mathcal{N}(0, 0.001)$. The Adam optimizer is used with learning rate $3e{-}4$ and the batch size was set to be 100. Finally, DirVAE sets $\boldsymbol{\alpha} = 0.98 \cdot \mathbf{1}_{50}$ by using Eq. (3).

6.2.2. Experiment results

Table 5 enumerates the performances of GVAE, SBVAE, and DirVAE, and the result shows the error rate of classification using 10%, 5% and 1% of labeled data for each dataset. We additionally used 25% of labeled data for SVHN and CIFAR-10 datasets which are more complex datasets than the MNISTs. For some cases in the MNIST dataset, which can be considered as rather simple tasks, GVAE shows the least error rate. However, in general, the experiment shows that DirVAE has the best performance out of three alternative VAEs. Also, it should be noted that the performance of DirVAE is more improved in the most complex task with the SVHN and CIFAR-10 dataset.

6.3. Application 3: Experiments of topic model augmentation with DirVAE

6.3.1. Experiment settings

One usefulness of the Dirichlet distribution is being a conjugate prior to the multinomial distribution, so it has been widely used in the field of topic modeling, such as LDA [1]. Recently, some neural variational topic (or document) models have been suggested, for example, ProdLDA [13], NVDM [32], and GSM [33]. NVDM used GVAE, and the GSM used GVAE-Softmax to make the sum-to-one positive topic vectors. Meanwhile, ProdLDA assume the prior distribution to be the Dirichlet distribution with the softmax Laplace approximation. To verify the usefulness of DirVAE, we replace the probabilistic encoder part of DirVAE to each model.

For the topic model augmentation experiment, two popular performance measures in the topic model fields, which are perplexity and topic coherence via normalized pointwise mutual information (NPMI) [34], have been used with *20Newsgroups*[1] and *RCV1-v2*[2] datasets. Perplexity is a function of document-wise averaged per-word log-likelihood which can be computed by $\exp(-\frac{1}{D}\sum_k \frac{\log p(\mathbf{x}_k)}{N_k})$ where D is the number of documents, N_k is the number of words in the kth document, and $\mathbf{x}_k$ is the kth document input. NPMI measures topic coherence by counting the co-occurence of words in the top-k words for each topic which can be computed by $-\log\frac{p(w_1,w_2)}{p(w_1)p(w_2)} \Big/ \log p(w_1, w_2)$. The lower is better for the perplexity, and the higher is better for the NPMI.

20Newsgroups has 11,258 train data and 7487 test data with vocabulary size 1995. For the RCV1-v2 dataset, due to the massive size of the whole data, we randomly sampled 20,000 train data and 10,000 test data with vocabulary size 10,000.

The specific model structures can be found in the original papers, Srivastava and Sutton [13], Miao et al. [32], 33]. We replace the model prior to that of DirVAE to each model and search the hyper-parameter as Table 6 with 1000 randomly selected test data. We use 500-dimension hidden layers and 50 topics for 20Newsgroups, and 1000-dimension hidden layers and 100 topics for RCV1-v2.

6.3.2. Experiment results

Table 7 indicates that the augmentation of DirVAE improves performance in general. Additionally, the best performers from the

[1] https://github.com/akashgit/autoencoding_vi_for_topic_models.
[2] http://scikit-learn.org/stable/datasets/rcv1.html.

two measurements are always the experiment cell with DirVAE augmentation except for the perplexity of RCV1-v2, which still remains competent. Table 8 shows top-10 high probability words per topic by activating single latent dimensions in the case of 20Newsgroups. Also, we visualized the latent embeddings of documents by t-SNE in Fig. 6.

7. Conclusion

Recent advances in VAEs have become one of the cornerstones in the field of DGMs. The VAEs infer the parameters of explicitly described latent variables, so the VAEs are easily included in the conventional PGMs. While this merit has motivated the diverse cases of merging the VAEs to the graphical models, we ask the fundamental quality of utilizing GVAE where many models have latent values to be categorical probabilities. The softmax function cannot reproduce the multi-modal distribution that the Dirichlet distribution can. Recognizing this problem, there have been some previous works that approximated the Dirichlet distribution in the VAE settings by utilizing the Weibull distribution or the softmax Gaussian distribution, but DirVAE with the inverse Gamma CDF shows better learning performance. Even though GVAE reconstructs the original input better than any other VAEs including DirVAE, DirVAE produces more meaningfully learned latent representation in our experiments: the supervised classifications, the semi-supervised, and the topic model augmentations. Moreover, DirVAE shows no component collapsing, and this property leads to better latent representations and performance gains. The proposed DirVAE can be widely used in the future in several perspectives: (1) due to the multi-modal property, DirVAE produces better latent representation than GVAE which is popularly used in the machine learning field, and hence it could replace GVAE when the task requires disentangled latent representations; and (2) if we recall the popularity of the conjugate relation between the multinomial and the Dirichlet distributions, DirVAE can be a brick to the construction of complex probabilistic models with neural networks.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was conducted at High-Speed Vehicle Research Center of KAIST with the support of Defense Acquisition Program Administration (DAPA) and Agency for Defense Development (ADD).

References

[1] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. (2003).
[2] W.K. Wong, Discover latent discriminant information for dimensionality reduction: non-negative sparseness preserving embedding, Pattern Recognit. 45 (2012) 1511–1523.
[3] C. Lee, E. Ahmed, T. Marwan, Learning representations from multiple manifolds, Pattern Recognit. 50 (2016) 74–87.
[4] D.P. Kingma, M. Welling, Auto-encoding variational bayes, in: International Conference on Learning Representations, 2014.
[5] X. Ye, J. Zhao, Multi-manifold clustering: a graph-constrained deep non-parametric method, Pattern Recognit. 93 (2019) 215–227.
[6] W. Xu, K. Shawn, G. Wang, Toward learning a unified many-to-many mapping for diverse image translation, Pattern Recognit. 93 (2019) 570–580.
[7] I. Goodfellow, P.-A. J., M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. (2014).
[8] E. Nalisnick, P. Smyth, Stick-breaking variational autoencoders, in: International Conference on Learning Representations, 2017.
[9] H. Ishwaran, L.F. James, Gibbs sampling methods for stick-breaking priors, J. Am. Stat. Assoc. (2001).
[10] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax, in: International Conference on Learning Representations, 2017.
[11] C.J. Maddison, A. Mnih, Y.W. Teh, The concrete distribution: a continuous relaxation of discrete random variables, in: International Conference on Learning Representations, 2017.
[12] J.T. Rolfe, Discrete variational autoencoders, in: International Conference on Learning Representations, 2017.
[13] A. Srivastava, C. Sutton, Autoencoding variational inference for topic models, in: International Conference on Learning Representations, 2017.
[14] M. Hoffman, M. Johnson, Elbo surgery: yet another way to carve up the variational evidence lower bound, Advanced in Neural Information Processing Systems Workshop in Approximate Bayesian Inference, 2016.
[15] J. Pitman, Combinatorial stochastic processes, Technical report, University of California, Berkeley, 2002.
[16] D.P. Kingma, M. Welling, Efficient gradient-based inference through transformations between bayes nets and neural nets, in: International Conference on Machine Learning, 2014.
[17] H. Zhang, B. Chen, D. Guo, M. Zhou, Whai: Weibull hybrid autoencoding inference for deep topic modeling, in: International Conference on Learning Representations, 2018.
[18] D.J.C. MacKay, Choice of basis for laplace approximation, Mach. Learn. (1998).
[19] D.A. Knowles, Stochastic gradient variational bayes for gamma approximating distributions, arXiv preprint arXiv:1509.01631(2015).
[20] T. Minka, Estimating a Dirichlet distribution, Technical Report, M.I.T., 2000.
[21] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. Inst. Electr.Electron. Eng. (1998).
[22] B.M. Lake, R.R. Salakhutdinov, J. Tenenbaum, One-shot learning by inverting a compositional causal process, Adv. Neural Inf. Process. Syst. (2013).
[23] S.A. Nene, S.K. Nayar, H. Murase, Columbia object image library, Technical Report, Columbia University, 1996.
[24] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A.Y. Ng, Reading digits in natural images with unsupervised feature learning, Advanced in sNeural Information Processing Systems Workshop on Deep Learning and Unsupervised Feature Learning, 2011.
[25] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Technical Report, University of Toronto, 2009.
[26] V. Nair, G. Hinton, Rectified linear units improve restricted Boltzmann machines, in: International Conference on Machine Learning, 2010.
[27] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: International Conference on Artificial Intelligence and Statistics, 2010.
[28] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980(2014).
[29] L.V.D. Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. (2008).
[30] Y. Burda, R. Grosse, R. Salakhutdinov, Importance weighted autoencoders, in: International Conference on Learning Representations, 2016.
[31] D.P. Kingma, S. Mohamed, D.J. Rezende, M. Welling, Semi-supervised learning with deep generative models, Adv. Neural Inf. Process. Syst. (2014).
[32] Y. Miao, L. Yu, P. Blunsom, Neural variational inference for text processing, in: International Conference on Machine Learning, 2016.
[33] Y. Miao, E. Grefenstette, P. Blunsom, Discovering discrete latent topics with neural variational inference, in: International Conference on Machine Learning, 2017.
[34] J.H. Lau, D. Newman, T. Baldwin, Machine reading tea leaves: automatically evaluating topic coherence and topic model quality, European Chapter of the Association for Computational Linguistics, 2014.

Weonyoung Joo received the BS and MS degrees from the Department of Mathematical Sciences, KAIST, in 2014 and 2016, respectively. He is currently working toward the doctoral degree in the Department of Industrial and Systems Engineering, KAIST.

Wonsung Lee received the Ph.D. degree in industrial and systems engineering from KAIST in 2018. He joined AI Technology Unit, SK Telecom, South Korea, as a data scientist.

Sungrae Park received the Ph.D. degree in industrial and systems engineering from KAIST in 2018. He is currently working for CLOVA AI Research Team, Naver corp., South Korea.

Il-Chul Moon is an associate professor at Dept. of Industrial and Systems Engineering, KAIST.