Approximate Inference for Deep Latent Gaussian Mixtures

Eric Nalisnick[1], **Lars Hertel**[2], and **Padhraic Smyth**[1]

[1]Department of Computer Science
[2]Department of Statistics
University of California, Irvine
{enalisni, lhertel, p.smyth}@uci.edu

1 Introduction

Deep latent Gaussian models (DLGMs) composed of density and inference networks [14]—the pipeline that defines a *Variational Autoencoder* [8]—have achieved notable success on tasks ranging from image modeling [3] to semi-supervised classification [6, 11]. However, the approximate posterior in these models is usually chosen to be a factorized Gaussian, thereby imposing strong constraints on the posterior form and its ability to represent the true posterior, which is often multimodal. Recent work has attempted to improve the quality of the posterior approximation by altering the *Stochastic Gradient Variational Bayes* (SGVB) optimization objective. Burda et al. [2] proposed an importance weighted objective, and Li and Turner [10] then generalized the importance sampling approach to a family of α-divergences. Yet, changing the optimization objective is not the only way to attenuate posterior restrictions. Instead, the posterior form itself can be made richer. For instance, Kingma et al. [7] employ full-covariance Gaussian posteriors, and Nalisnick & Smyth [13] use (truncated) GEM random variables. This paper continues this later line of work by using a Gaussian mixture latent space. We describe learning and inference for not only the traditional mixture model but also Dirichlet Process mixtures [1] (with posterior truncation). Our *deep Latent Gaussian mixture model* (DLGMM) generalizes previous work such as Factor Mixture Analysis [12] and Deep Gaussian Mixtures [15] to arbitrary differentiable inter-layer transformations.

2 Latent Gaussian Mixtures

We now describe a novel modification of the DLGM/VAE in which we use a Gaussian mixture model (GMM) as the approximate posterior. We modify the generative process to be $\boldsymbol{\pi}_i \sim \text{Dir}(\boldsymbol{\alpha})$, $\mathbf{z}_i \sim \sum_{k=1}^{K} \pi_{i,k} N(\mathbf{z}; \boldsymbol{\theta}_k)$, $\mathbf{x}_i \sim p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_i)$ where $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_i)$ is the density network. We assume the posterior factorizes as $q(\boldsymbol{\pi}, \mathbf{z}|\mathbf{x}_i) = q(\boldsymbol{\pi}|\mathbf{x}_i)q(\mathbf{z}|\boldsymbol{\pi}_i, \mathbf{x}_i) = \prod_{K-1} \text{Kumar}(a_{i,k}, b_{i,k}) \sum_{k=1}^{K} \pi_{i,k} N_{\boldsymbol{\theta}_k}(z|\mathbf{x}_i)$ where Kumar(a,b) denotes the Kumaraswamy distribution [4]. Notice that we bypass the complication of sampling valid mixture weights $\boldsymbol{\pi}_i$ by, firstly, using the Dirichlet's marginal (aka 'stick-breaking') construction and, secondly, employing the Kumaraswamy as the approximate posterior for the Dirichlet's marginal Betas. The Kumaraswamy has a closed-form inverse CDF that can serve as a valid *differentiable non-centered parametrization* (DNCP) [13] whereas the Beta has no such DNCP. Having defined the prior and posterior, we now can write the SGVB evidence lowerbound (ELBO) for this model as:

$$\begin{aligned}\mathcal{L}_{\text{SGVB}} = \sum_{k} \mu_{\pi_k} [\frac{1}{S} \sum_{s} \log p_{\boldsymbol{\theta}}(\mathbf{x}_i|\hat{\mathbf{z}}_{i,k,s}) + \mathbb{E}_{q_k}[\log p(\mathbf{z}_i)]] \\ - \text{KLD}[q(\boldsymbol{\pi}_k|\mathbf{x}_i)||p(\boldsymbol{\pi}_k)] - \frac{1}{S} \sum_{s} \log \sum_{k} \hat{\pi}_{i,k,s} q(\hat{\mathbf{z}}_{i,k,s}; \boldsymbol{\phi}_k)\end{aligned} \tag{1}$$

Workshop on Bayesian Deep Learning, NIPS 2016, Barcelona, Spain.

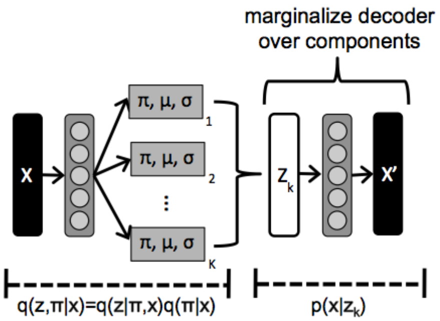
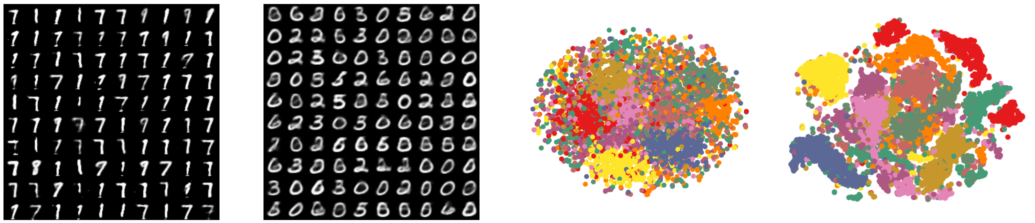


Figure 1: Computation graph of a *Deep Latent Gaussian Mixture Model* (DLGMM). The inference network computes the parameters of K mixture components. The decoder network receives a sample from each and computes the reconstruction. The recursive process by which the mixture weights π_k are generated is omitted.



(a) $\mu = -1.5$ (b) $\mu = 1.5$ (c) Single Gaussian (d) Gaussian Mixture (K=5)

Figure 2: Subfigures (a) and (b) show samples from the two mixture components at the extremes of the latent space. Subfigures (c) and (d) show t-SNE embeddings of the Gauss-VAE and DLGMM latent space (respectively).

where $\hat{\pi}$ and $\hat{z}$ are S samples taken via non-centered parametrizations and μ_{π_k} is the mean of the posterior weight distribution. This model has the benefit of relatively straightforward DNCPs but has the drawback of needing to run the density network ('decoder') K times, where K is the number of components, for each forward pass. This expensive marginalization is required because of the difficulty in sampling from the mixture directly, i.e. $\mathbf{z} \sim \sum_k \pi_k q_k(\mathbf{z})$[1].

The computation path of the the proposed DLGMM is summarized in Figure 1. The inference network computes the parameters of the K mixture components, and the density network is run for a sample from each. The mixture weight, once sampled, is used no where in the computation path to reconstruct the data. Rather its influence is in the ELBO, weighting each term according to the corresponding component. Equation 1 can be extended to multiple stochastic layers, but the density network must be run K^s times, where K is the number of components and s the number of stochastic layers, for each forward pass.

As we are already using the Dirichlet's stick-breaking construction, it is easy to extend the model to infinite mixtures defined by the Dirichlet Process (assuming posterior truncation), i.e. $G(\cdot) = \sum_{k=1}^{\infty} \pi_k \delta_{\zeta_k}$ where δ_{ζ_k} is a discrete measure concentrated at $\zeta_k \sim G_0$ and the π_ks are, again, random weights chosen independent of G_0 such that $0 \leq \pi_k \leq 1$ and $\sum_k \pi_k = 1$. The only significant change is the prior on the Beta marginals. For all k, we have $v_{i,k} \sim \text{Beta}(1, \alpha_0)$ where α_0 is the concentration parameter. We assume the variational posterior takes the same form as above and is truncated to T components, as is usually done when performing variational inference for DP mixtures [1].

3 Experiments

We compared our proposed *deep latent Gaussian mixture model* (DLGMM) and *deep latent Dirichlet Process mixture model* (DLDPMM) to the single-Gaussian VAE/DLGM (Gauss-VAE) [8, 14] and the *stick-breaking* VAE (SB-VAE) [13] on the binarized MNIST dataset and Omniglot [9], using the pre-defined train/valid/test splits. We optimized all models using AdaM [5] with a learning rate of 0.0003 (other parameters kept at their Tensorflow defaults), batch sizes of 100, and early stopping

[1] Alex Graves' note *Stochastic Backpropagation through Mixture Density Distributions* describes a technique for calculating gradients though samples from a mixture model, but we found the method requires many samples (100+) of the latent variables and did not result in models with competitive marginal likelihoods.

	k=3	k=5	k=10
DLGMM	$\mathbf{9.14}$	$\mathbf{8.38}$	$\mathbf{8.42}$
SB-VAE	9.34	8.65	8.90
Gauss-VAE	28.4	20.96	15.33

(a) MNIST test error for kNN on latent space

	$-\log p_{\boldsymbol{\theta}}(\mathbf{x}_i)$	
	MNIST	OMNIGLOT
DLGMM (500d-3x25s)	$\mathbf{96.50}$	123.50
DLDPMM (500d-17tx25s)	96.91	123.76
Gauss-VAE (500d-25s)	96.80	$\mathbf{119.18}$
SB-VAE (500d-25t)	98.01	–

(b) Estimated Marginal Likelihood

Figure 3: Subtable (a) shows MNIST test error for kNN classifiers trained on samples from the latent distributions. Results for 3, 5, and 10 (k) neighbors are given. Each model was trained with no label supervision. Subfigure (b) reports the (Monte Carlo) estimated marginal likelihood on the test set.

with 30 look-ahead epochs. For the marginal likelihood results, all Gaussian priors are standard Normals and all Dirichlets are symmetric, with $\alpha = 1$, except for the DLDPMM, which has $\alpha_0 = 1$.

Qualitative evaluation. First we compared the models qualitatively by examining samples and class distribution within the latent space. Samples from two components of a 5-component DLGMM are shown in Subfigures (a) and (b) of Figure 2. Normal priors were placed on the five components with means set to $\mu = \{-1.5, -.75, 0, .75, 1.5\}$ and all variances set to one. The samples are from the extremes of the prior, i.e. $\mu_1 = -1.5$ and $\mu_5 = 1.5$. We see that the DLGMM learned not only recognizable MNIST digits but also to divide their factors of variation into different parts of the latent space. Thin digits such as sevens are generated from the component with the most negative prior mean and wide digits such as zeros are generated from the component with the most positive prior mean. Also we visually examined the MNIST class distribution in the latent space via t-SNE projection. The 2D embeddings are shown in Figures 2 (c) and (d) for the Gauss-VAE and DLGMM respectively; colors denote digit classes. The DLGMM's latent space exhibits conspicuously better clustering. We validate this observation empirically below using kNN.

Quantitative evaluation. We compared the models quantitatively using a k-Nearest Neighbors (kNN) classifier on their latent space as well as by calculating the marginal likelihood of a held-out set. Table (a) of Figure 3 reports MNIST test error for kNN classifiers trained on the latent space of the Gauss-VAE, a SB-VAE, and the proposed DLGMM. Note that none of these models had access to labels during training. We see from the table that the DLGMM performs markedly better than the Gauss-VAE—supporting our visual analysis above of the t-SNE projections—and slightly better than the SB-VAE. Moreover, the DLGMM's superior performance holds across all number of neighbors tested ($k = \{3, 5, 10\}$).

Lastly, in Figure 3 (b) we report the (Monte Carlo) estimated marginal likelihood for the various models on MNIST and Omniglot. The network architectures are given in parentheses: *d* denotes a deterministic layer, *s* a stochastic layer, $K\times$ the number of mixture components, and *t* the truncation level for the DP and SB models. We find that using a mixture latent space improves the likelihood modestly for MNIST but not at all for Omniglot.

4 Conclusions

In this paper we extended the DLGM/VAE to mixture latent spaces and proposed solutions—such as using a stick-breaking construction and the Kumaraswamy for the marginal distribution of the mixture weights—to the complications with learning and inference in this class of deep generative model. Furthermore, our innovations support multiple stochastic layers as well as infinite mixtures (with a truncated variational approximation); however, the cost of marginalizing the decoder can become prohibitive in these cases. We experimentally compared the DLGMM to the single Gaussian and Stick-Breaking VAEs and found that, intuitively, the mixture latent space provides better clustering into the data's natural structure (such as MNIST digit style and class).

References

[1] David M Blei and Michael I Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143, 2006.

[2] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *International Conference on Learning Representations*, 2016.

[3] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1462–1471, 2015.

[4] MC Jones. Kumaraswamy's distribution: A beta-type distribution with some tractability advantages. *Statistical Methodology*, 6(1):70–81, 2009.

[5] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

[6] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[7] Diederik P. Kingma, Tim Salimans, and Max Welling. Improving variational inference with inverse autoregressive flow. *Neural information processing systems*, 2016.

[8] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *International Conference on Learning Representations*, 2014.

[9] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[10] Yingzhen Li and Richard E Turner. Renyi divergence variational inference. *Neural information processing systems*, 2016.

[11] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. *International Conference on Machine Learning*, 2016.

[12] Angela Montanari and Cinzia Viroli. Heteroscedastic factor mixture analysis. *Statistical Modelling*, 10(4):441–460, 2010.

[13] Eric Nalisnick and Padhraic Smyth. Nonparametric deep generative models with stick-breaking priors. *ICML Workshop on Data-Efficient Machine Learning*, 2016.

[14] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of The 31st International Conference on Machine Learning*, pages 1278–1286, 2014.

[15] Aaron van den Oord and Benjamin Schrauwen. Factoring variations in natural images with deep gaussian mixture models. In *Advances in Neural Information Processing Systems*, pages 3518–3526, 2014.