JMLR: Workshop and Conference Proceedings 1–6

Normalized Random Measure Mixture Models in Variational Autoencoders

Rahul Mehta MEHTA5@UIC.EDU and **Hui Lu** HUILU@UIC.EDU
Department of Bioengineering
University of Illinois at Chicago

1. Introduction

Generating interpretable latent information from data sets is a major challenge that is addressed by various popular methods such as topic models, latent feature allocation, and mixture models. As data sets increase in size and complexity the models are limited by computational power. There are a variety techniques to alleviate the computational burden such as collapsed Gibbs sampling (CGS) and collapsed variational Bayes (VB) inference. For BNP models the advantage of CGS is the unbounded latent space, amassing new clusters is organically folded into the sampling steps. The drawback is the increasing number of parameters that must be stored on the order of the number for each observation. Although this is potentially solved by using parallelization schemes, the computational complexity remains linear with the data, which is still inefficient for large data sets. With VB and stochastic variational inference (SVI) the model converges more efficiently at the cost of introducing a truncation scheme, a contradictory approach for nonparametric models.

A promising approach to address computational tractability and truncation are variational autoencoders (VAE) (Kingma and Welling, 2013) where the generative model is learned jointly alongside the inference scheme. This is intuitively appealing because the latent space is parameterized by the neural network such that the observations are mapped to a distribution over latent variables. In contrast to SVI that optimizes every data point, amortized inference in VAE models avoids the need to update individual parameters for all observations allowing for efficient test-time inference as it only requires one forward pass through the neural network for new data. For example, there is an ongoing effort to obtain disentangled features in VAEs as in (Nalisnick and Smyth, 2016)(Goyal et al.), where the priors are based on a stick breaking process and the nested Chinese restaurant process respectively, however, to solve the optimization problem both models required truncated versions of the priors. Other proposed priors are based on Gaussian Mixture models (GMM) (Dilokthanakul et al., 2016), but requires specifying the size of the data set a priori.

The aim of this paper is not to outperform state-of-the-art approaches but rather to create a more expressive representation of large datasets while still remaining computationally tractable. The NRM prior is inefficient on large datasets, but the clusters it produces are useful for creating a clear and descriptive picture from the data. VAEs, on the other hand, are useful with larger datasets but the latent space is typically restricted by Gaussian models. The combination of both VAE and NRM mixture models gives the best of both worlds: greater application by stochastic dimensionality and a richer latent representation. Our paper seeks to elucidate how to implement the framework for incorporating an NRM mixture model as a prior for VAE. This, in turn, helps to show how control the latent representation of the data with specific parameters without setting an implicit truncation level.

2. Background

A completely random measure (CRM) (Kingman, 1967), is a random measure u on Ω, such that for any disjoint collection of sets, A_k, the corresponding measure of those sets are independent random variables identified by mass, π_k and location θ_k. To construct a mixture model with a CRM, it is normalized by its finite mass, T, to create the normalized random measure $\tilde{u} = \Sigma_{k=1}^{\infty} \frac{\pi_k}{T} \delta_{\theta_k}$ Normalized random measures encompass many of the models in Bayesian nonparametrics, such as the Dirichlet process, which is a normalization of the gamma process. We can define our NRM mixture model as

$$u \sim CRM(\rho, \mathbb{H}_0) \quad \tilde{u} \sim \frac{u}{T} \quad z_k|\tilde{u} \sim \text{Discrete}(\tilde{u}), \quad x_k|z_k, \theta_k \ \sim \text{Multi}(\cdot|\theta_{z_k}) \tag{1}$$

where z_k indexes which component x_k belongs to, θ_k are the components of the mixture model drawn i.i.d from a base distribution $H_0(\cdot)$ (in our case a Dirichlet), and the last term is the observation model (a multinomial distribution).

The levy measure, ρ, is the fundamental tool that defines the discrete non-parametric prior. For example, the underlying levy measure of the Dirichlet process has a Gamma intensity $\rho_a(ds) = as^{-1}e^{-s}$ where a is the concentration parameter. To increase flexibility in the mixture model, we choose the normalized generalized Gamma process (NGGP) (Favaro et al., 2013) that is parameterised by the levy intensity $\rho_{\alpha,\sigma,\tau}(\pi) = \frac{\alpha}{\Gamma(1-\sigma)} s^{-\sigma-1} e^{-\tau\pi}$ where $\alpha > 0, \sigma \in (0, 1)\text{and}\tau \geq 0$. With specific parameters the NGGP can create the DP when $\rho_{\alpha,0,0}$, the $\sigma-$stable process given by $\rho_{\alpha,\sigma,0}$,and the normalized inverse Gaussian process, given by $\rho_{\alpha,\frac{1}{2},\tau}$. The NGGP follows a Chinese restaurant process update rule by marginalizing out the weights (James et al., 2009) and conditioning on an auxiliary variable $U_n \sim \Gamma(n, T)$.

3. Variational Autoencoders for the NGGP Mixture Model

By assuming a conjugate base measure, the distribution of NGGP with the weights marginalized out is

$$p(z_{1:n}, u, \theta, x_{1:n}) \propto \frac{u^{n-1}}{\Gamma(n)} \prod_{k=1}^{\infty} \kappa_{n_k}(u) \prod_{z_n = k} F(x_n|\theta_k) H(\theta_k) \tag{2}$$

where $\kappa_{n_k}(u)$ denotes the n_kth moment of the exponentially tilted Levy measure (see (Favaro et al., 2013) for the formulation). The ELBO is then written as

$$\begin{aligned} \mathcal{L}_{ELBO} = \mathbb{E}_{q_{(\mathbf{z},\theta|x)}}[\ln p_\rho(x, \mathbf{z}, \theta)] + \mathbb{E}_{q_{(\mathbf{z}|x)}}[\ln q_\beta(z|x)] + \mathbb{E}_{q_{(\theta|x)}}[\ln q_\phi(\theta|x)] + \\ \mathbb{E}_{q_{(u|\mathbf{z},x)}}[\ln q_\gamma(u|\mathbf{z}, x)] \end{aligned} \tag{3}$$

where ρ, β, ϕ, and γ are the neural network parameters. The individual ELBO terms are called the reconstruction error and the entropies of $\theta, \mathbf{z}, u$ respectively. As the reparameterization trick is only applicable to cases where the latent variables are differentiable, non-centered parameteriation (DNCP), we can extend it a variety of distributions using acceptance rejection sampling (Naesseth et al., 2017) given the fundamental lemma

Lemma 1 (Lemma 1) *Let f be any measurable function and $\epsilon \sim \pi(\epsilon|\mu) = s(\epsilon)\frac{g(h(\epsilon,\mu|\mu))}{r(h(\epsilon,\mu|\mu))}$ the distribution of the accepted sample. Then:*

$$\mathbb{E}_{\pi(\epsilon|\mu)}[f(h(\epsilon,\mu))] = \int f(h(\epsilon,\mu))\pi(\epsilon|\mu)d\epsilon = \int f(w)g(w|\mu)dw = \mathbb{E}_{g(w|\mu)}[f(w)] \tag{4}$$

Then the gradient can be taken using the log derivative trick:

$$\nabla_\mu \mathbb{E}_{\pi(\epsilon|\mu)}[f(h(\epsilon,\mu))] = \mathbb{E}_{\pi(\epsilon|\mu)}[\nabla_\mu f(h(\epsilon,\mu))] + \mathbb{E}_{\pi(\epsilon|\mu)}\left[f(h(\epsilon,\mu))\nabla_\mu \log\frac{g(h(\epsilon,\mu|\mu))}{r(h(\epsilon,\mu|\mu))}\right] \tag{5}$$

If we assume the approximate posterior takes the form $g(w|\mu)$, the above lemma allows us to instead reparameterize the proposal distribution $r(w|\mu)$ where $w = h(\epsilon,\mu)$ and $\epsilon \sim s(\epsilon)$.

The reconstruction error and the entropy of $q(\theta)$ both follow Dirichlet distributions with concentration parameters $\alpha_{1:k}$. Using θ as an example for the formulation, we can see that if $\widetilde{\theta}$ = Gamma(α_k,1) i.i.d, then $\theta_{1:K} = (\sum_l \widetilde{\theta}_l)^{-1}(\widetilde{\theta_1}, ..., \widetilde{\theta_K})^T \sim Dirichlet(\alpha_{1:K})$. Therefore we can use **Lemma 1** for reparameterization,

$$\mathbb{E}_{q(\theta_{1:K}|\alpha_{1:K})}[f(\theta_{1:K})] = \int f\Big(\frac{\bar{\theta}_{1:K}}{\sum_l^K \widetilde{\theta}_l}\Big)\prod_{k=1}^{K} Gamma(\bar{\theta}_k;\alpha_k,1)d\bar{\theta}_{1:K} \tag{6}$$

Furthermore the transformed gamma distributed variables are simulated as $\bar{\theta}_k = h_{Gamma}(\epsilon_k,\alpha_k) = (\alpha_k - \frac{1}{3})(1 + \frac{\epsilon}{\sqrt{9\alpha - 1}})^3 \text{where} \epsilon_k \sim \mathcal{N}(0,1)$.

For $q(z|x)$ we follow (Kurihara et al.) and approximate using $\exp(\mathbb{E}_{q_{\theta_k}}[\ln f(x_n|\theta_k)]) + \mathbb{E}_{q_{z_{-n}}}[\ln p(z|z_{-n})]$. The first term is the reconstruction error and the second term can be represented as $\mathbb{E}[\ln p(z|z_{-n})] = \Sigma_{z_{-n}}\prod_{j\neq n} q(z_j)\ln p(z_n = k|z_j)$, which is approximated by the first order Taylor expansion.

$$\mathbb{E}[\ln p(z|z_{-n})] \propto \begin{cases} \log \mathbb{E}[n_{-k}] - \sigma + \log(1 - \mathrm{Var}[n_{-k}]), & \text{if k < K.} \\ \log \alpha(U_n - \tau)^\sigma, & \text{if k = } \varnothing. \end{cases} \tag{7}$$

where n_k is the amount of observations in cluster k and z_{-n} are all the other clusters. We can see that the z-posterior calculates clustering assignment probability directly from a scaled reconstruction term.

For $q(u|z)$, we represented u as a log-concave function with a change in variables $V = \log(U)$ as in (Favaro et al., 2013)

$$q(V|z) \propto \frac{e^{vn}}{(e^v + \tau)^{n - \alpha\,\mathbb{E}[q(\mathbf{z})]}} e^{\frac{\alpha}{\sigma}((e^v+\tau)^\sigma - \tau^\sigma)} \tag{8}$$

where $\mathbb{E}[q(\mathbf{z})]$ is the expected number of clusters. At the cost of high variance, we use the score function gradient as it is applicable to any variational distribution. We hypothesize, the tension between the posterior-z term and the reconstruction error will minimize the variance as $q(z|x)$ captures the relationship between u and $\mathbf{z}$.

4. Experiments

To validate our approach we perform clustering on the Fashion MINST data set due to its larger variety of possible latent embeddings. Architecture, hyperparameter, and figures are detailed in Appendix B. We We train a VAE with the NGGP prior defined in (1) and evaluate its clustering performance on the test data set. Our initial cluster size is 5 and is dynamically grown based on the probability of accepting a new cluster until convergence. To compare cluster assignments we use the protocol created by (Makhzani et al., 2015)] where we find the element of the test set with the highest probability of belonging to cluster k and assign that label to all other test samples belonging to k. This is then repeated for all clusters and the assigned labels are compared with the true labels to obtain an unsupervised classification error rate. A summary of the results obtained on the Fashion MINST benchmark is shown in Table 1. We achieve classification scores competitive with approaches also modifying the VAE prior. Empirically, we observe allowing the model to determine the amount of clusters helps achieve better performance, reinforcing the hypothesis that setting a truncation level leads to a loss of information.

We test the generative capabilities of our model by generating samples for any specified class as in Figure 1 comparing our model with the top-performing InfoGan. Qualitative assessment shows comparable results where some classes for InfoGAN are more distinct, while our model over-generates some classes. We suspect some clusters have too many over-lapping features and need to be merged.

5. Conclusion

Our main goal was to explore how we can scale a NRM prior to larger datasets while remaining computationally tractable. To do so we incorporated the NGGP mixture model as a prior in a VAE. Our algorithm leverages the amortized inference of a VAE while maintaining the infinite-dimensional nature and clustering prowess of the NGGP mixture model. The key to tractability was use of the accept-rejection reparameterization sampler as it updates three of the four terms in the loss function. Another point to note, is that we marginalized out the weights, however this is a significant drawback as it limits the model to use conjugate priors. A possible way to solve this, is to use slice sampling for the weights or adaptive thinning as proposed in the original MCMC characterization (Favaro et al., 2013) of the NGGP mixture model. The computational power required depends on if we can restrict the weights to be sampled locally. We also suspect we can improve performance by finding a better variational distribution for $q(u|z)$ or a score function with lower variance. Another natural question is the inclusion of data-driven split-merge moves, as it could improve the generative capabilities of our model. The current presentation indicates there are various promising areas of future work and further research can improve the flexibility of both the NGGP mixture model and VAEs.

References

Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

Stefano Favaro, Yee Whye Teh, et al. Mcmc for normalized random measure mixture models. *Statistical Science*, 28(3):335–359, 2013.

Prasoon Goyal, Zhiting Hu, Xiaodan Liang, Chenyu Wang, and Eric P Xing. Nonparametric variational auto-encoders for hierarchical representation learning.

Lancelot F James, Antonio Lijoi, and Igor Prünster. Posterior analysis for normalized random measures with independent increments. *Scandinavian Journal of Statistics*, 36 (1):76–97, 2009.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

John Kingman. Completely random measures. *Pacific Journal of Mathematics*, 21(1):59–78, 1967.

Kenichi Kurihara, Max Welling, and Yee Whye Teh. Collapsed variational dirichlet process mixture models.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Christian Naesseth, Francisco Ruiz, Scott Linderman, and David Blei. Reparameterization gradients through acceptance-rejection sampling algorithms. In *Artificial Intelligence and Statistics*, pages 489–498, 2017.

Eric Nalisnick and Padhraic Smyth. Stick-breaking variational autoencoders. *arXiv preprint arXiv:1605.06197*, 2016.

Appendix A. Gradient Calculation

In our case we must find the gradients for $\mathbb{E}_{q_\rho(\theta,z,u|x;\alpha,\sigma,\tau)}[\log p_\phi(x|z,\theta,u)]$, ρ are the parameters of the NGGP. As specified in Section 3 we can use Lemma 1 and optimize the unbiased Monte Carlo estimates of the gradient, such that the gradient is:

$$\nabla_\phi \mathbb{E}_{q_\rho(\theta,z,u|x;\alpha,\sigma,\tau)}[\log p_\phi(x|z,\theta,u) = g_{rep} + g_{cor} \tag{9}$$

$$g_{rep} = \nabla_\rho \log p_\phi(x, h(\epsilon,\alpha)) = \nabla_\theta \log p(x,\theta) \nabla_\alpha h(\epsilon,\alpha) \tag{10}$$

where $\nabla_\alpha h(\epsilon,\alpha) \propto (\alpha - 1 + x) * \frac{1}{Dirichlet(\alpha)}$ and the Dirichlet is generated from the transformation mentioned in Section 3. We can see that g_{rep} is the gradient of the reconstruction loss with respect to α and can be handled using automatic differentiation packages.

$$g_{cor} = \nabla_\alpha \log q(h(\epsilon,\alpha)) + \nabla_\alpha \log \frac{dh}{d\epsilon}(\epsilon,\alpha) \tag{11}$$

$$\nabla_\alpha \log q(h(\epsilon,\alpha)) = \log h(\epsilon,\alpha) + (\alpha - 1) + \frac{\frac{dh(\epsilon,\alpha)}{d\alpha}}{h(\epsilon,\alpha)} - \frac{dh(\epsilon,\alpha)}{d\alpha} - \psi(\alpha) \tag{12}$$

Appendix B. Experimental Details

B.1. Architecture and Hyperparameters

For both the encoder and decoder we use MLP with 2 hidden layers of size [400 200] and [200 400] hidden units respectively. We trained until convergence using early-stopping with a look ahead of 30 epochs. We used the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 1e-3, and mini-batches of size 64.

Method	**K**	**Result**
SB-VAE (Nalisnick and Smyth, 2016)	20	85.09
GMVAE (Dilokthanakul et al., 2016)	20	77.78
NGGP	20	82.31
NGGP	32	90.45
NGGP	40	81.53

Table 1: Unsupervised classification accuracy of Fashion MINST with various cluster sizes

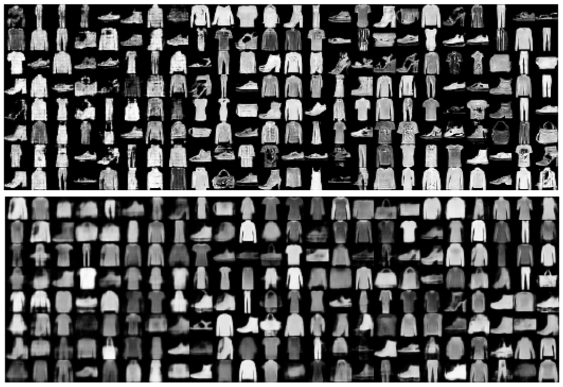
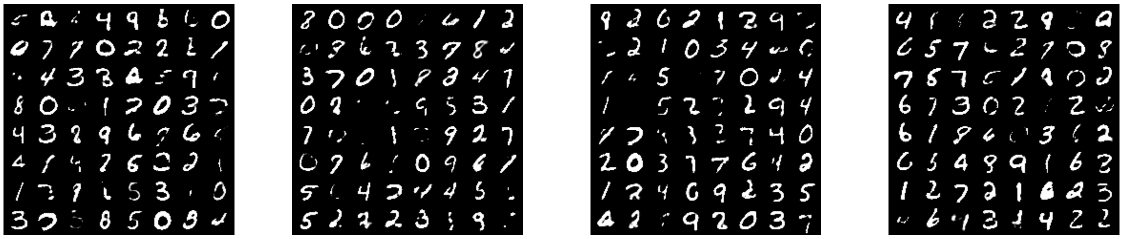


Figure 1: Generated samples from Fashion MINST, top is our model, bottom is InfoGAN



(a) k = 2 (b) k = 4 (c) k = 7 (d) k = 10

Figure 2: Random samples from our model of MNIST for different dimensionalities of latent space.