RICE UNIVERSITY
STAT 525
Bayesian Statistics
Alex Ho

Final Project Report:
Variational Autoencoders for Approximate Posterior Inference

1 Summary

With large or complex data, there are many instances where the posterior distribution of the data is difficult or impossible to compute. *"Autoencoding Variational Bayes"* by Diederik P. Kingma and Max Welling [6] proposes the Autoencoding Variational Bayes algorithm and the Variational Autoencoder to allow for approximate inference of the posterior distribution under circumstances where traditional Markov Chain Monte Carlo methods are not sufficient. In this paper, we detail the motivation and formulation for variational inference methods (based on *"Variational Inference: A Review for Statisticians"* by David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe [2]), as well as how [6] acts as a potential approach to variational inference. Additionally, the proposed model and algorithm are evaluated on two image data sets to determine if it is a useful for approximate Bayesian inference.

2 Introduction

Bayesian methods depend on the use of probability distributions to conduct inference on data by the computation of a posterior distribution. Modern statistics, however, often use data that do not conform to standard distributions, making posterior distributions difficult or even impossible to compute. As such, a significant portion of Bayesian research has been dedicated to developing algorithms to approximate such posterior densities. A popular and effective approach to approximate posterior computation has been the Markov chain Monte Carlo (MCMC) sampling method with the Metropolis-Hastings algorithm and Gibbs sampling. Nevertheless, there are still cases under which MCMC methods are not sufficient when faster and less computationally expensive solutions are needed, as with large data

sets and very complex models. Thus, variational inference (VI) methods were developed to compute approximate posterior densities by optimization as opposed to sampling. While VI methods do not have the same desirable asymptotic guarantees that MCMC methods have, they often are "close enough" to the target density and have faster computations to warrant their use in a few different cases.

The Variational Autoencoder (VAE) model and the Autoencoding Variational Bayes (AEVB) algorithm were first proposed by [6] in 2013 as an approach to VI. Although it was proposed as a VI method, it soon became a popular method in the machine learning and deep learning communities as a generative model for complex data. In machine learning, the applications of the VAE focus on the concept that the latent space is an efficient and compact representation of the data, rather than the traditional Bayesian interpretation of approximating posteriors. Now, there are many variations of the VAE model which can handle sequential and spatial data, with applications in independent component analysis, disentangled representation learning, and deep sample generation. While the VAE model has proven useful in these deep learning applications, its use as a VI technique is less prominent. This project will explore the effectiveness of the VAE model and its AEVB algorithm from the Bayesian perspective for VI and modern data analysis.

3 Overview

3.1 The Challenge of Intractable Posteriors and MCMC

We will consider the following scenario. Let $X = \{x^{(i)}\}_{i=1}^{N}$ be an independent and identically distributed sample of some arbitrary variable x, whose random generation process is dependent on an unobserved continuous latent variable z. More precisely, each $z^{(i)}$ is generated by a prior distribution $p_{\theta^*}(z)$, and each $x^{(i)}$ is generated by a conditional distribution $p_{\theta^*}(x|z)$, each coming from a parametric distribution family $p_\theta(z)$ and $p_\theta(x|z)$ respectively whose probability density function is differentiable almost everywhere with respect to the true parameter θ^* and z. Notably, there are no additional simplifying assumptions about the marginal or posterior probabilities. Given this scenario, we would like to estimate the

posterior $p_{\theta*}(z|x)$ from the data. However there is one key challenge: when calculating the marginal likelihood $p_\theta(x) = \int p_\theta(x|z)p_\theta(z)dz$, the latent variables are by nature unobserved, may have an a large dimension, and the integral may not have a closed form analytical solution, so this computation is often intractable.

Fortunately, there have been methods from Markov chain Monte Carlo sampling which grants us approximate posterior inference. The Metropolis-Hastings algorithm [8] and the Gibbs sampler [3] are two of the most widely used forms of approximate inference. However, there are still situations where these methods are not useful. For instance, if we have large data sets, methods like Monte Carlo Expectation-Maximization are too computationally expensive because it requires an entire sampling loop per data point.

3.2 Variational Inference

In instances where there are large data sets or the model is too complicated, variational inference methods offer a computationally faster and scalable solution to inference. Variational inference methods use optimization to approximate the posterior within a family of proposed distributions $\mathcal{Q}$. The goal is to find a member of the family q which minimizes the Kullback-Leibler (KL) divergence with the exact posterior $q^*(z) = \text{argmin}_{q(z)\in\mathcal{Q}} KL(q(z) \parallel p(z|x))$. The original inference problem is now framed as an optimization problem, and the complexity of the proposed distribution family, which is chosen by the user, is the main contributor to the complexity of this problem.

However, the KL-divergence takes the following form

$$\begin{aligned} KL(q(z) \parallel p(z|x)) &= \mathbb{E}\left[\log q(z)\right] - \mathbb{E}\left[\log p(z|x)\right] \\ &= \mathbb{E}\left[\log q(z)\right] - \mathbb{E}\left[\log p(z,x)\right] + \log p(x) \end{aligned}$$

which shows its dependence on $p(x)$, the nuisance term in our original problem. Thus, we must use an objective function which is equivalent to the KL-divergence up to an additive

constant, the Evidence Lower BOund, or ELBO, which takes the following form:

$$\begin{aligned} ELBO(q) &= \mathbb{E}[\log p(z,x)] - \mathbb{E}[q(z)] \\ &= \mathbb{E}[\log p(z)] + \mathbb{E}[\log p(x|z)] - \mathbb{E}[\log q(z)] \\ &= \mathbb{E}[\log p(x|z)] - KL(q(z) \parallel p(z,x) \end{aligned}$$

Maximizing this objective function encourages the distribution q to place mass where the expected log likelihood is increased, meaning that the latent variables explain the observed data well, and decreased mass where the difference between q and the prior $p(z)$ is large.

Another observation to note about the ELBO is that it lower bounds the log evidence. Given the fact that the KL-divergence is always non negative, along with previous equations, we can see that

$$\begin{aligned} \log p(x) &= KL(q(z) \parallel p(z|x)) + ELBO(q) \\ \log p(x) &\geq ELBO(q) \end{aligned}$$

Since this bound is a good approximation for the marginal likelihood, it gives us a reasonable criterion for selecting a good model, although it is not necessarily correct.

Now that we have a proper objective function to optimize, we need a model of families to choose from. One simple family for VI is the mean-field variational family. In this family, the latent variables are assumed to be mutually independent each with their own variational factor. A member of the mean-field variational family may take the following form:

$$q(z) = \prod_{j=1}^{m} q_j(z_j)$$

We can notice that the mean-field variational family does not directly depend on the data x, but rather only through the maximization of the ELBO and the minimization of the KL-divergence are the data and model linked together. Each individual variational factor does

not need to take any specific form, and is usually chosen to be appropriate given the desired posterior. As mentioned before, the complexity of optimizing the mean-field variational family primarily depends on the choice of the distribution family, and the trade-off between goodness of fit and the difficulty of the optimization can be determined by the user and the situational need. Although the mean-field family can be useful, it makes a strong assumption that the variational factors are mutually independent, so there will be a poor fit if the true posterior contains correlations.

One of the most popular algorithms to solve the mean-field variational family optimization is coordinate ascent variational inference (CAVI) shown in [1]. CAVI approaches a local optimum by holding all mean-field variational density factors constant except one, and iteratively optimizing for the unfixed variational density factor. The optimal $q_j(z_j)$ keeping all other variational factors $q_\ell(z_\ell),\ \ell \neq j$ constant, is proportional to the exponentiated expected log of the complete conditional,

$$q_j^*(z_j) \propto \exp\{\mathbb{E}_{-j}[\log p(z_j|z_{-j}, x)]\}$$
$$q_j^*(z_j) \propto \exp\{\mathbb{E}_{-j}[\log p(z_j, z_{-j}, x)]\}$$

and we can see the second proportionality holds because each latent variable and variational factor are mutually independent. Thus, the core of the CAVI algorithm is to iteratively set each $q_j(z_j)$ to be proportional to $\exp\{\mathbb{E}_{-j}[\log p(z_j|z_{-j}, x)]\}$ until the ELBO converges.

Although the CAVI algorithm with the mean-field variational family is a simple yet effective method for VI, it still has weaknesses. In particular, the algorithm does not easily scale to large data sets. This weakness is due to the fact that for each iteration of the algorithm, the entire data set needs to be iterated over. Thus, with large modern data sets which may have millions of samples, it is not computationally reasonable to use the CAVI algorithm.

In order to overcome scaling problems, gradient-based optimization is useful. Stochastic

Variational Inference (SVI) [4] combines natural gradients with stochastic optimization by using noisy yet cheap gradient computations to reach local maximums of the ELBO. SVI optimizes the global parameters by maintaining the current estimates for the model, minibatch sampling from the full data set, computing the optimal local parameters using the minibatch and current global parameters, and adjusting the global parameters, repeating until convergence. This method of stochastic optimization is the same method which allowed [7] to perform neural network-based image recognition for tens of thousands of images.

This use of cheap and unbiased natural gradients allow for easy scaling, and has led to several models and algorithms taking advantage of such conveniences. One proposal to the SVI and learning algorithm is Autoencoding Variational Bayes and the Variational Autoencoder model, which we will now discuss in detail.

3.3 Autoencoding Variational Bayes Algorithm

Diederik P. Kingma and Max Welling first proposed the AEVB algorithm in [6] in order to answer the following question: "How can we perform efficient approximate inference and learning with directed probabilistic models whose continuous latent variables and/or parameters have intractable posterior distributions?"

By deriving a new lower bound estimator, the AEVB aims to allow maximum likelihood (ML) or maximum a posteriori (MAP) inference with the global parameters and VI on the latent variables. We will still consider the problem stated in Section 3.1. Let the recognition model $q_\phi(z|x)$ be an approximation of the true posterior $p_\theta(z|x)$. This recognition model does not take any closed analytical form, but rather ϕ is learned jointly with the generative parameters θ. The same variational lower bound we derived before can also be written as

$$\mathcal{L}(\theta, \phi; x^{(i)}) = -KL(q_\phi(z|x^{(i)}) \parallel p_\theta(z)) + \mathbb{E}_{q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)}|z)\right]$$

however, the naïve Monte Carlo gradient estimator, the authors note, has very high variance. To solve this problem, the authors propose a reparamaterization of the random variable

$\tilde{z} \sim q_\phi(z|x)$ to be a differentiable transformation $g_\phi(x, \epsilon)$ where $\epsilon \sim p(\epsilon)$ is an auxiliary noise variable. Now form Monte Carlo estimates of expectations of some function $f(z)$ with respect to $q_\phi(z|x)$ by the following:

$$\mathbb{E}_{q_\phi(z|x^{(i)}}\left[f(z)\right] = \mathbb{E}_{p(\epsilon)}\left[f(g_\phi(\epsilon, x^{(i)}))\right] \simeq \frac{1}{L}\sum_{l=1}^{L} f(g_\phi(\epsilon^{(l)}, x^{(i)}))$$

where $\epsilon^{(l)} \sim p(\epsilon)$, which allows us to have a generic Stochastic Gradient Variational Bayes (SGVB) estimator $\mathcal{L}(\theta, \phi; x^{(i)}) \simeq \tilde{\mathcal{L}}^A(\theta, \phi; x^{(i)}) = \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(x^{(i)}, z^{(i,l)}) - \log q_\phi(z^{(i,l)}|x^{(i)}))$ where $z^{(i,l)} = g_\phi(\epsilon^{(i,l)}, x^{(i)})$ and $\epsilon^{(l)} \sim p(\epsilon)$ when applying it to the lower bound.

The KL divergence can be computed analytically in some cases, so the authors note we can estimate the expected reconstruction error $\mathbb{E}_{q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)}|z)\right]$ by sampling. This gives us another version of the SGVB estimator:

$$\tilde{\mathcal{L}}^B(\theta, \phi; x^{(i)}) = -KL(q_\phi(z|x^{(i)}\|p_\theta(z)) + \frac{1}{L}\sum_{l=1}^{L}(\log p_\theta(x^{(i)}|z^{(i,l)})$$

where again we have $z^{(i,l)} = g_\phi(\epsilon^{(i,l)}, x^{(i)})$ and $\epsilon^{(l)} \sim p(\epsilon)$.

By observing the above estimator, we can note that the KL divergence between the approximated posterior and the prior acts as a regularizer for the negative expected reconstruction error. This interpretation gives us a natural formulation for this estimator to be implemented with auto-encoders as auto encoders aim to reconstruct a given input from a learned compact representation. In this case, we use $g_\phi(.)$ to map a data point $x^{(i)}$ with random noise vector $\epsilon^{(l)}$ to the approximate point estimate for the posterior for that data point. We can then input a sample $z^{(i,l)}$ from the chosen prior distribution to a learned likelihood function to output the probability density of $x^{(i)}$ under the generative model given the sample.

In order to use the solution above as formulated, we need to choose a deterministic reparamaterization function $g_\phi(.)$ so that $z = g_\phi(\epsilon, x)$. Here we will consider the location-

scale family of distributions (as we will use the Gaussian distribution in our implementation). Let the auxiliary variable ϵ be sampled from a standard distribution (location = 0 and scale = 1). Then let $g(.) = \textit{location} + \textit{scale} \cdot \epsilon$. For example, let $z \sim p(z|x) = \mathcal{N}(\mu, \sigma^2)$. The appropriate reparamaterization in this case would be $z = \mu + \sigma\epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$.

With all these parts in place, the final algorithm acts as other SVI methods: keeping track of the global estimates for θ and ϕ, sample a minibatch drawn from the full data set; sample from the auxiliary noise distribution; compute the gradient of the minibatch estimator based on the drawn minibatch; update the parameters based on the gradient accordingly.

3.4 The Variational Autoencoder Model

Now let us consider an implementation of the AEVB algorithm with the Variational Autoencoder. Let the approximation of the true posterior be the encoder $q_\phi(z|x)$ made of a neural network, the prior be a centered isotropic Gaussian $P_\theta(z) = \mathcal{N}(z; 0, I)$ of the appropriate dimension, and the likelihood $p_\theta(x|z)$ be a multivariate Gaussian or Bernoulli (depending on the data) with distribution parameters learned from z with a neural network, and the variational approximate posterior be a multivariate Gaussian with diagonal covariance $\log q_\phi(z|x^{(i)}) = \log \mathcal{N}(z; \mu^{(i)}, \sigma^{2(i)} I)$. Since the prior $p_\theta(z)$ and the approximate posterior $q_\phi(z|x)$ are Gaussian, we can use $\tilde{\mathcal{L}}^B(\theta, \phi; x^{(i)})$ as the estimator and the KL divergence can be computed and differentiated without sampling. Thus, the closed form estimator (which can be fully differentiated and computed) is the following:

$$\mathcal{L}(\theta, \phi; x^{(i)}) \simeq \frac{1}{2} \sum_{j=1}^{J} \left(1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2\right) + \frac{1}{L} \sum_{l=1}^{L} \log p_\theta(x^{(i)}|z^{(i,l)})$$

where $z^{(i,l)} = \mu^{(i)} + \sigma^{(i)} \odot \epsilon^l$ and $\epsilon^{(l)} \sim \mathcal{N}(0, I)$. This model formulation is the approximate model which we will experiment with in the following section.

4 Extensions

4.1 Experiments

VAEs are typically used in machine learning applications with image data, such as with the MNIST data set [7] in the original proposal. Therefore, for the experiments, we will be replicating the paper's results with the MNIST data set as well as applying it to the Fashion MNIST data set [9]. Fashion MNIST is a dataset images with a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. MNIST is a dataset of handwritten digits with the same training set and test set size as Fashion MNIST.

4.1.1 Methods

All experiments were done in Google Colab with the TensorFlow and Keras libraries. The code for the analysis on MNIST can be found here, while the code for the Fashion MNIST analysis can be found here. The implementations were based on TensorFlow tutorials for Convolutional Variational Autoencoders, the Keras tutorial for VAEs, and the TFP Probabilistic Layers: Variational Auto Encoder tutorial. While there is no official code available from the authors of [6], these implementations are standard examples introducing the VAE and only have minor differences from the what was described in the original proposed model. The primary differences are that we do not use the same nonlinearities described in their multilayer perceptron, and as we are using image data, we are using a convolutional neural network [7] as opposed to the assumed multilayer perceptron in the paper. These changes should have no impact on the end results because both the multilayer perceptron and convolutional neural networks are universal approximators [10].

Each model was based on standard convolutional neural network architectures. The VAE with standard layers was trained over 30 epochs (30 passes over the data set) while the VAE with probabalistic layers was trained over 15 epochs. Both used the default TensorFlow hyperparameters for the Adam optimizer [5]. Both the probabalistic model and the standard

model had a latent dimension of 2 (2 means and 2 variances estimated).

Data was minimally processed. The value of each pixel was set to zero or one depending on whether the pixel value was less than or greater than the median value (255/2).

4.1.2 Notes

Traditional Bayesian inference involves drawing from the posterior distribution, and then estimating credible intervals and posterior predictive distributions to compare to the original data. However, the standard VAE model only produces point estimates, and there is no way to directly sample from the learned posterior distribution. The best approximation we can get to drawing from the posterior distribution is by inputting unseen testing data to the VAE and plotting the point estimates from the posterior.

Despite the difficulty of sampling from posterior predictive distribution, it is fairly simple to draw from the prior predictive distribution because of the use of the reparamaterization trick. Thus, we can at least evaluate the learned likelihood distribution by drawing from the standard normal distribution and using it as an input to the decoder. Although, visual inspection alone of these prior predictive samples gives a weak indication that the model is learning the likelihood and the posterior in a reasonable way beyond just maximizing the ELBO.

Moreover, it is difficult to give a direct comparison to other traditional posterior approximation techniques. Although the paper details the assumptions for the likelihood and prior, it is impossible to replicate such assumptions in other frameworks, even black-box models like Stan or JAGS because we cannot provide the assumption that true model parameters are the output of the latent variables run through a multilayer perceptron.

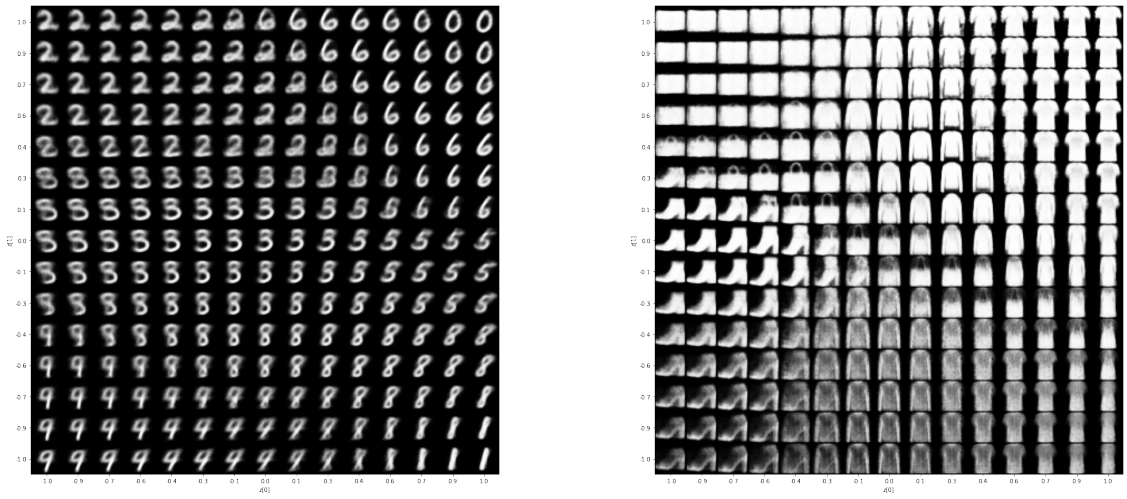
4.2 Results

To evaluate the VAE model for Bayesian inference, we conducted four different checks after training on two data sets: (1) traversing the prior predictive sample space; (2) posterior predictive point estimates on a standard model; (3) posterior predictive point estimates on a probabalistic model; (4) plotting the posterior means of the latent variables.

The original paper used the MNIST data, however the data are quite uniform in structure and do not have wide variability. To further test how the VAE handles complex data, we evaluated the model on the Fashion MNIST data set.

Additionally, the standard VAE implementation only has deterministic layers, which does not allow for uncertainty in the outputs. To allow for uncertainty in the reconstructions, a probabalistic VAE was also implemented so that the probability of the output pixels being present is a Bernoulli variable. This allowed us to view different aspects of the posterior predictive point estimates, which we will discuss.

4.2.1 Prior Predictive Sample Space



(a) MNIST (b) Fashion MNIST

Figure 1: Prior predictive sample space traversal

After the standard VAE was trained, uniformly spaces samples were drawn from the prior distribution and then fed to the decoder. Recalling the model formulation, the decoder acts as the learned likelihood $p(x|z)$. The goal of this evaluation was to determine whether the prior predictive samples were reasonably recovering the inputs, which we can see in Figure 1 that it does a reasonable job at generating recognizable samples.

(a) Original Fashion MNIST testing data sample

(b) Posterior predictive point estimate sample

Figure 2: Posterior predictive point estimates for Fashion MNIST with the standard model.

(a) Original MNIST testing data sample

(b) Posterior predictive point estimate sample

Figure 3: Posterior predictive point estimates for MNIST with the standard model.

4.2.2 Posterior Predictive Point Estimates: Standard VAE

Because the standard VAE model cannot sample directly from the posterior distribution, the closest we can do to evaluating the posterior is through the posterior predictive point estimates. By using points taken from the out of sample testing data, we can approximate posterior point estimates, which are then fed through the decoder to recover posterior predictive point estimates. We can see that the recovered images look very similar to the original samples, however the edges of each digit are not well defined. See Figure 2 and Figure 3 for more details.

4.2.3 Posterior Predictive Point Estimates: Probabalistic VAE

For this evaluation, we followed the same process to get posterior predictive point estimates, however the VAE's output layer was a Bernoulli distribution for the probability that each pixel was active. We can see that the posterior predictive modes are very similar to the original input, and the variances are high (white) at the edges of each number, similar to

what we noticed before, while they are low for the background and center of the numbers. See Figure 4 and Figure 5 for more details.

(a) Original MNIST testing data sample

(b) Posterior predictive point estimate sample

(c) Mode

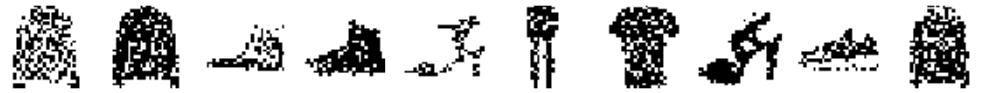
(d) Mean

(e) Variance

Figure 4: Posterior predictive point estimates for MNIST with the probabalistic model. For the sample, mode, mean, and variance, each pixel has probability of being colored represented by a Bernoulli distribution.

4.2.4 Posterior Means

Finally, we look to the posterior means in Figures 6 and 7. We created these plots by saving the posterior mean point estimates for each sample. Here, the points for each latent variable mean are colored by label. For both the MNIST and Fashion MNIST data sets, the test data groups are grouped together in the same way as the training data, but more sparsely. This indicates that the model handles testing data well and maps it correctly to the posterior mean.



(a) Original Fashion MNIST testing data sample

(b) Posterior predictive point estimate sample

(c) Mode

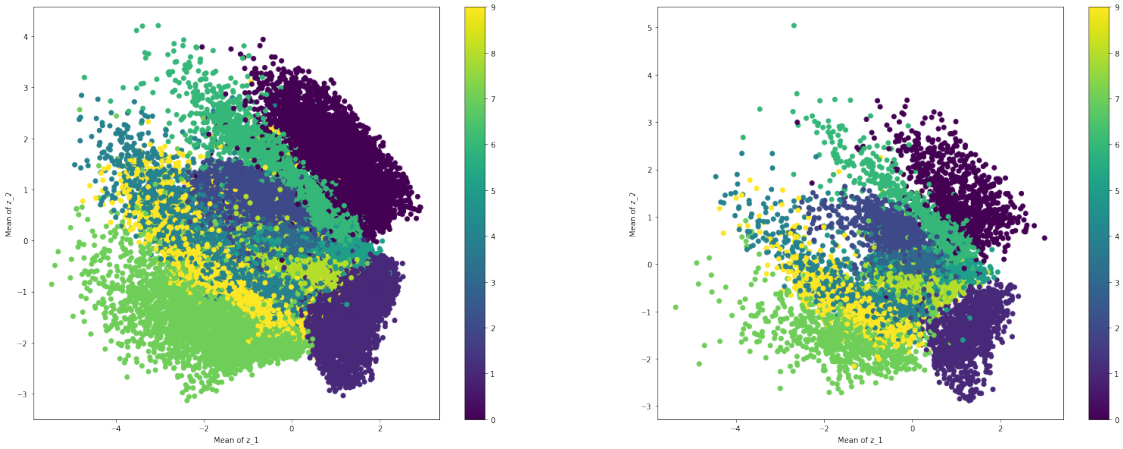
(d) Mean

(e) Variance

Figure 5: Posterior predictive point estimates for Fashion MNIST with the probabalistic model. For the sample, mode, mean, and variance, each pixel has probability of being colored represented by a Bernoulli distribution.

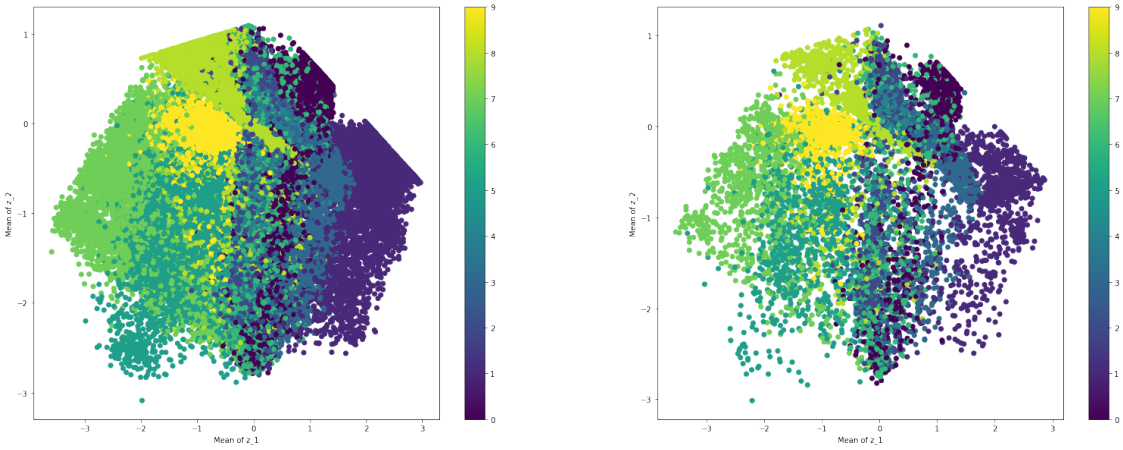
5 Conclusion

In conclusion, we demonstrated how VAEs can be used for Bayesian inference. While it acts as a good point estimator for MAP and ML, it does not allow for standard posterior checks by drawing from the posterior. The VAE model does perform well as a standard autoencoder, it does not have the same asymptotic guarantees that MCMC methods have, nor the posterior inference evaluation techniques as other posterior approximates. The results support the idea that the data are well organized, but there are better VI methods to choose from.



(a) Train data (b) Test data

Figure 6: Scatter plot of posterior means for MNIST latent variables



(a) Train data (b) Test data

Figure 7: Scatter plot of posterior means for Fashion MNIST latent variables

References

[1] Christopher M Bishop. "Pattern recognition". In: *Machine learning* 128.9 (2006).

[2] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. "Variational inference: A review for statisticians". In: *Journal of the American statistical Association* 112.518 (2017), pp. 859–877.

[3] Stuart Geman and Donald Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images". In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1984), pp. 721–741.

[4] Matthew D Hoffman et al. "Stochastic variational inference." In: *Journal of Machine Learning Research* 14.5 (2013).

[5] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[6] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[7] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[8] Nicholas Metropolis et al. "Equation of state calculations by fast computing machines". In: *The journal of chemical physics* 21.6 (1953), pp. 1087–1092.

[9] Han Xiao, Kashif Rasul, and Roland Vollgraf. "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms". In: *arXiv preprint arXiv:1708.07747* (2017).

[10] Ding-Xuan Zhou. "Universality of deep convolutional neural networks". In: *Applied and computational harmonic analysis* 48.2 (2020), pp. 787–794.