Seeing the whole picture instead of a single point: Self-supervised likelihood learning for deep generative models

Petra Poklukar POKLUKAR@KTH.SE
Judith Bütepage BUTEPAGE@KTH.SE
Danica Kragic DANI@KTH.SE
KTH Royal Institute of Technology, Stockholm, Sweden

Abstract

Recent findings show that deep generative models can judge out-of-distribution samples as more likely than those drawn from the same distribution as the training data. In this work, we focus on variational autoencoders (VAEs) and address the problem of misaligned likelihood estimates on image data. We develop a novel likelihood function that is based not only on the parameters returned by the VAE but also on the features of the data learned in a self-supervised fashion. In this way, the model additionally captures the semantic information that is disregarded by the usual VAE likelihood function. We demonstrate the improvements in reliability of the estimates with experiments on the FashionMNIST and MNIST datasets.

Keywords: Variational Autoencoders, Semantic Likelihood, Self-Supervised Learning

1. Introduction

Deep Generative Models (DGMs) have gained in popularity due to their ability to model the density of the observed training data from which one can draw novel samples. However, as Nalisnick et al. (2018) pointed out in their recent paper, the inferences made by likelihood-based models, such as Variational Autoencoders (VAEs) (Kingma and Welling, 2015; Rezende et al., 2014) and flow-based models (Kingma and Dhariwal, 2018; van den Oord et al., 2016), are not always reliable. They can judge out-of-distribution (OOD) samples to be more likely than in-distribution (ID) samples that are drawn from the same distribution as the training data. Concretely, a DGM trained on the FashionMNIST dataset will on average assign higher likelihoods to images from the MNIST dataset than to test images from the FashionMNIST dataset (see for example top left image in Figure 1(a)).

In this work we tackle the problem of misaligned likelihood estimates produced by VAEs on image data and propose a novel likelihood estimation during test time. Our method leverages findings reported in our earlier work Bütepage et al. (2019), which are summarised in Section 2, and is based on the idea to evaluate a given test image not only locally, using individual parameters returned by a VAE as it is usually done, but also globally using learned feature representations of the data. The main contribution of this paper is the introduction of a feature-based likelihood trained in a self-supervised fashion. This likelihood evaluates the model also based on the semantics of a given image and not solely on the values of each pixel. We elaborate on this idea in Section 3 and demonstrate the improvements with an empirical evaluation presented in Section 4. We emphasise that the aim of our work is exclusively to improve the reliability of the likelihood estimation produced by VAEs. We focus on image data in particular as we have not observed the misalignment in our earlier experiments on various non-image datasets from UCI Machine Learning Repository (Dua and Graff, 2017). We plan to investigate this further in the future work. Due to the lack of space we omit the experiments on non-image data as well as the specifics of VAEs for which we refer the reader to Kingma and Welling (2015); Rezende et al. (2014).

2. Modeling and evaluation assumptions influencing likelihood estimates

This section provides a background on the evaluation of VAEs and summarizes our earlier work presented in (Bütepage et al., 2019).

In VAEs, the observed random variable $\mathbf{X}$ is assumed to be generated from the joint distribution $p(\mathbf{X}, \mathbf{Z}) = p(\mathbf{X}|\mathbf{Z})p(\mathbf{Z})$ where $\mathbf{Z}$ denotes the latent variables. Using variational inference the intractable true posterior distribution $p^*(\mathbf{Z}|\mathbf{X})$ is approximated with a simpler parametrised distribution $q(\mathbf{Z}|\mathbf{X})$. VAEs employ amortized inference where encoder and decoder neural networks, $\phi_z(\mathbf{X})$ and $\phi_x(\mathbf{Z})$, are jointly trained to represent the approximate posterior distribution $q(\mathbf{Z}|\phi_z(\mathbf{X}))$ and likelihood function $p(\mathbf{X}|\phi_x(\mathbf{Z}))$, respectively.

From a Bayesian perspective, we can evaluate a successfully trained VAE using two different evaluation schemes

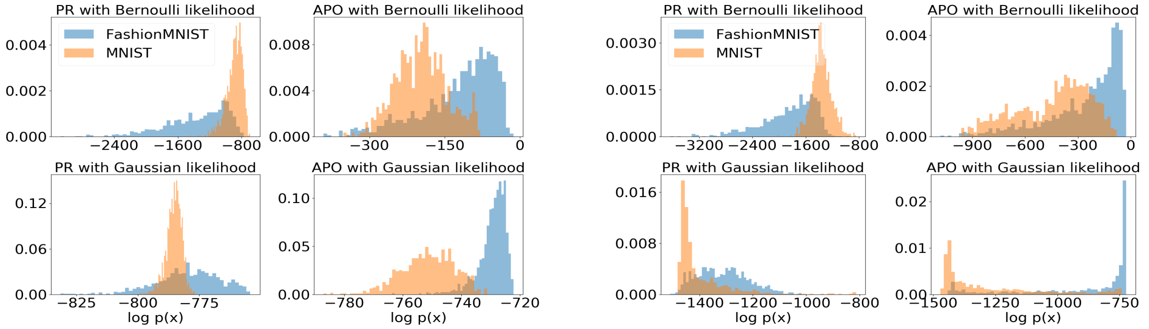
$$p_{VAE}^{PR}(\mathbf{x}|\mathbf{X}) = \int_{\mathbf{z}} p(\mathbf{x}|\phi_x(\mathbf{z}))p(\mathbf{z}) \tag{1}$$

$$p_{VAE}^{APO}(\mathbf{x}|\mathbf{X}) = \int_{\mathbf{z}} p(\mathbf{x}|\phi_x(\mathbf{z}))q(\mathbf{z}|\phi_z(\mathbf{x})), \tag{2}$$

where p_{VAE}^{PR} denotes the prior predictive (PR) and p_{VAE}^{APO} the approximate posterior predictive (APO) distribution. Bütepage et al. (2019) argue that the likelihood estimates produced by a trained VAE are influenced by both 1) the choice of the above listed evaluation scheme and 2) the choice of the parametrisation of the likelihood function $p(\mathbf{X}|\phi_x(\mathbf{Z}))$. Here, two common choices are a Gaussian distribution in the case of colored images or a Bernoulli distribution in the case of black and white (or grey-scaled) images. The effect of both 1) and 2) is best demonstrated in Figure 1(*a*) where we visualise the log likelihood estimates from a VAE V_1, parametrised by a Bernoulli likelihood (top row), and a VAE V_2, parametrised by a Gaussian likelihood (bottom row), using both PR (left column) and APO (right column) evaluation schemes from Equations (1) and (2). Both VAEs were trained on the FashionMNIST dataset and tested on test images from both the FashionMNIST and MNIST datasets. In the case of V_1 the pixel values of the images were binarised with threshold 0.5, and in the case of V_2 scaled to the interval $[0, 1]$.

The choice of the evaluation scheme influences the variance of the estimates of the training data as it directly affects the variability of the parameters $\phi_x(\mathbf{z})$ returned by the VAE (see left vs right column in Figure 1(*a*)). Namely, PR produces more diverse parameters corresponding to the latent representations of the whole training data while APO generates more homogeneous samples corresponding to the latent representation of a given test point $\mathbf{x}$. On the other hand, the choice of the likelihood parametrisation (top vs bottom row in

Figure 1(a)) influences the actual values of the estimates since images are evaluated under distributions of different shapes. We refer the interested reader to (Bütepage et al., 2019) for a detailed discussion. Note that only the top-left combination in Figure 1(a) reproduces the results reported in (Nalisnick et al., 2018).



(a) Using the usual VAE likelihood $p_{VAE}(\mathbf{x}|\phi_x(\mathbf{z}))$.

(b) Using the improved $p_{FEVAE}(\mathbf{x}|\phi_x(\mathbf{z}))$ likelihood from Equation (4).

Figure 1: Normalised histogram of the log likelihood under the prior (left) and approximate posterior (right) using model V_1 with an iid Bernoulli likelihood function (top) and model V_2 with an iid Gaussian likelihood function (bottom).

3. Self-supervised likelihood learning

This section describes the self-supervised feature-based likelihood function which is the main contribution of this work.

In addition to the influencing factors discussed in Section 2, we hypothesise that the likelihood estimates are also affected by the assumption that image pixels are independent and identically distributed (iid) around the likelihood function parameterised by the decoder. Let a test image $\mathbf{x}$ be represented as a concatenated vector of length D and let $\mathbf{x}^d$ denote its d-th component. Using the assumption of iid pixels, the likelihood function becomes a product of individual pixel-wise likelihoods: $p_{VAE}(\mathbf{x}|\phi_x(\mathbf{z})) = \prod_{d=1}^{D} p(\mathbf{x}^d|\phi_x(\mathbf{z})^d)$. Therefore, when computing the probability of $\mathbf{x}$, the likelihood only captures pixel-wise errors that are evaluated *locally* under the parameters $\phi_x(\mathbf{z})$ returned by the VAE and does not take into account the *"global"* information contained in the image (such as the semantics of the dataset). To mitigate the lack of the global evaluation, we propose to weight the likelihood term during test time with an additional term that relates the semantic information of both the test point $\mathbf{x}$ and the parameters $\phi_x(\mathbf{z})$ to the semantics of the whole training dataset. We define the details below.

We separately train a self-supervised classifier Γ and use its l-th layer to extract a low dimensional feature representation $\mathbf{f}_x = l(\mathbf{x})$ of an image $\mathbf{x}$. We train Γ on the same training dataset $\hat{\mathbf{X}} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ as we train the VAE. We then fit a Bayesian Gaussian Mixture (BGM) model with C components to the set $\mathbf{F} = \{\mathbf{f}_{x_1}, \dots, \mathbf{f}_{x_n}\}$ of feature representations extracted from a randomly sampled subset of $\hat{\mathbf{X}}$ of size $n < N$ (see also Section 4 for details).

Let $\mathbf{f}_x$ be the feature representation of a test image $\mathbf{x}$. During the evaluation of the BGM on $\mathbf{f}_x$ each mixture component is assigned a weight that indicates its contribution to the generation of $\mathbf{f}_x$. Let C_x denote the mixture component with the highest weight. Given likelihood parameters $\phi_x(\mathbf{z})$ returned by the VAE, we define the *global likelihood* of $\mathbf{x}$ as the product

$$p_{FE}(\mathbf{f}_x|\mathbf{f}_{\phi_{\mathbf{x}}(\mathbf{z})}) = p_{FE}(\mathbf{f}_x|C_{\phi_x(\mathbf{z})})p_{FE}(\mathbf{f}_{\phi_x(\mathbf{z})}|C_{\phi_x(\mathbf{z})}) \tag{3}$$

where $p_{FE}(\mathbf{f}_x|C_{\phi_x(\mathbf{z})})$ is the likelihood of the test point in feature space under the mixture component $C_{\phi_x(\mathbf{z})}$ determined by the representation $\mathbf{f}_{\phi_x(\mathbf{z})}$ of the parameters $\phi_x(\mathbf{z})$ and $p_{FE}(\mathbf{f}_{\phi_x(\mathbf{z})}|C_{\phi_x(\mathbf{z})})$ is the likelihood of $\mathbf{f}_{\phi_x(\mathbf{z})}$ under the same component $C_{\phi_x(\mathbf{z})}$. The first term can be seen as a *global* likelihood of the test point under the decoded parameters and the second term represents a *global* likelihood of the parameters themselves.

We then propose to evaluate the test image $\mathbf{x}$ under the combined likelihood function

$$\begin{aligned} p_{FEVAE}(\mathbf{x}|\phi_x(\mathbf{z})) &:= p_{VAE}(\mathbf{x}|\phi_x(\mathbf{z}))p_{FE}(\mathbf{f}_x|\mathbf{f}_{\phi_{\mathbf{x}}(\mathbf{z})}) \qquad (4)\\ &= \left[\prod_{d=1}^{D} p(\mathbf{x}^d|\phi_x(\mathbf{z})^d)\right] p_{FE}(\mathbf{f}_x|\mathbf{f}_{\phi_{\mathbf{x}}(\mathbf{z})}) \end{aligned}$$

where p_{VAE} as before captures *local* pixel-wise errors and p_{FE} additionally captures the *global* (semantic) likelihood.

4. Experiments

We evaluate our method with experiments on FashionMNIST and MNIST datasets and present the results below.

Feature extraction We obtained low dimensional features of the training data by deploying a self-supervised Jigsaw classifier Γ presented by Noroozi and Favaro (2016). The classifier receives a Jigsaw puzzle, which is a shuffled 3×3 grid of tiles extracted from a given image, and outputs (the class of) the permutation that was applied to the original unshuffled grid (see Appendix A for the implementation details). Note that any self-supervised learning strategy could be deployed as long as the obtained low dimensional features are of high quality and represent the training data well. After the completed training we randomly sampled $n = 10000$ training images $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and obtained their low dimensional representations $\{\mathbf{f}_1, \ldots, \mathbf{f}_n\}$ from the first layer l^1 of the classifier Γ, to which we fitted a BGM model with $C = 15$ components. The parameters n and C were determined using a hyperparameter grid search. We used representations from the first layer because we hypothesise that the earlier layers of the classifier carry useful information about the training data while the later layers carry information about the task itself. We leave experiments with representations obtained from different layers for the future work.

Experiment We trained two VAEs, V_1 and V_2, and two Jigsaw classifiers, Γ_1 and Γ_2 with specifications described in Appendix A on the FashionMNIST dataset. Here, the subscripts 1 and 2 denote that the model in consideration was trained on images binarised with threshold 0.5 and on images with pixel values scaled to the interval $[0, 1]$, respectively. As in the experiment producing the results in Figure 1(a), V_1 additionally assumes a Bernoulli likelihood and V_2 a Gaussian likelihood. For a given (binarised) test image $\mathbf{x}$ and parameters

$\phi_x(\mathbf{z})$ obtained from the trained VAE V_i, we first calculate the VAE likelihood p_{VAE} in the usual way using the assumption of iid pixels. We then obtain their low dimensional features $\mathbf{f}_x = l_i^1(\mathbf{x})$ and $\mathbf{f}_{\phi_x(\mathbf{z})} = l_i^1(\phi_x(\mathbf{z}))$ from the first layer l_i^1 of the trained Jigsaw classifier Γ_i and calculate p_{FE} under the fitted BGM following Equation (3). The product of the two likelihoods then equals the newly proposed likelihood p_{FEVAE} from Equation (4).

Given this pipeline, $\text{VAE}_i + \Gamma_i$ for $i = 1, 2$ and our likelihood p_{FEVAE}, we compared the log likelihood estimates using the PR and APO evaluation schemes from Equations (1) and (2) on the images from the test splits of FashionMNIST and MNIST datasets. The results are visualised in Figure 1(b). We see that our method significantly improves the estimates when using Gaussian likelihood parametrisation (bottom row) as it clearly separates the OOD samples from the ID samples. Note that the VAE parameters $\phi_x(\mathbf{z})$ in the PR evaluation always reflect the distribution of the entire training data. This means that the global likelihood of a test point evaluates the test point under all classes that were presented during training time. In practice this means that the PR evaluation of the global likelihood averages over all classes which results in a less distinct separation of the OOD samples. When using Bernoulli likelihood (top row) our method increases the variance of the likelihood of OOD samples but fails to achieve the same separation as in the Gaussian case. This is because a significant amount of the semantic information is lost during the binarisation process of the FashionMNIST dataset. The resulting binarised images are often unrecognisable with a sparse pixel distribution which makes the task of solving Jigsaw puzzles more difficult. Since digits in MNIST images are also sparse they become likely under p_{FE}. We observe their estimates fusing with FashionMNIST estimates if we corrupt the background using salt and pepper noise (see Figure 2 in Appendix B). We therefore hypothesise that in this particular case OOD samples simply become too similar to the ID samples, suggesting that the Bernoulli likelihood is not the most appropriate modelling choice. The inadequacy of the Bernoulli distribution in VAEs has also recently been discussed by Loaiza-Ganem and Cunningham (2019) who instead suggest to use their fully characterized continuous Bernoulli distribution.

5. Conclusion

We have discussed how the problematic assumption that the image pixels are iid around the decoded parameters narrows the focus of the VAE likelihood function p_{VAE} to a local area of the data density. Thus, the model likelihood function disregards the global data density, including the semantic information. Our proposed likelihood function mitigates this problem by leveraging self-supervised feature learning. In the future, we aim to evaluate our method on more complex datasets, such as CIFAR-10 and SVHN, and to design an end-to-end training procedure of VAEs using our proposed likelihood.

References

Judith Bütepage, Petra Poklukar, and Danica Kragic. Modeling assumptions and evaluation schemes: On the assessment of deep latent variable models. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations (ICLR)*, 2015.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10236–10245, 2018.

Gabriel Loaiza-Ganem and John P Cunningham. The continuous bernoulli: fixing a pervasive error in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 13266–13276, 2019.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do Deep Generative Models Know What They Don't Know? *arXiv e-prints*, art. arXiv:1810.09136, Oct 2018.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286, 2014.

Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional Image Generation with PixelCNN Decoders. *arXiv e-prints*, art. arXiv:1606.05328, Jun 2016.

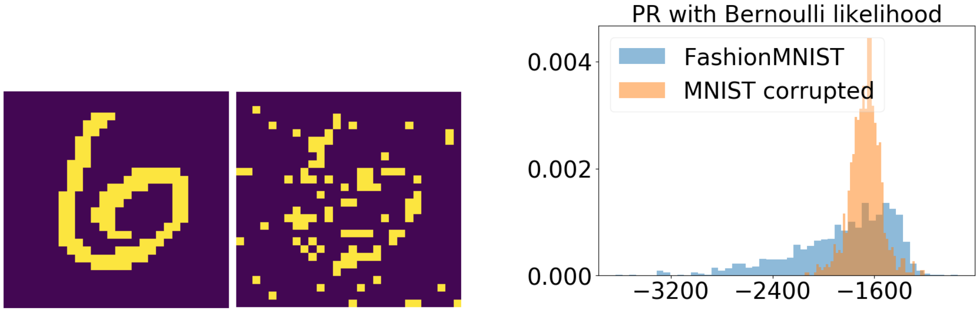
Appendix A. Implementation details

Feature extraction We designed the Jigsaw classifier as an MLP with four hidden layers of dimensions $100, 80, 60, 40$ respectively, each followed by a ReLU activation function and a dropout layer with rate 0.2. We extracted tiles of dimension 7×7 and generated 100 permutations. We trained the Jigsaw classifier for 200 epochs using SGD optimizer with 0.9 momentum, weight decay $5e-4$, learning rate 0.1 and batch size 128.

VAE For the VAEs we followed the setup used by Nalisnick et al. (2018) as closely as possible. We designed an encoder with five convolutional layers with feature maps of dimensions $8, 16, 32, 64, 64$ and stride $2, 1, 2, 1, 2$ respectively. The kernel size was set to 5 in all layers. The last convolutional layer was additionally followed by a linear layer of dimension 50. The decoder consisted of a linear layer of dimension 3136 and four transpose convolutional layers with feature maps of dimensions $32, 32, 3, 1$, kernel sizes $3, 3, 4, 5$ and stride $2, 1, 2, 1$ respectively. Each of the (transpose) convolutional layers was followed by the ReLU activation function, batch normalisation and a dropout layer with rate 0.1. We additionally used Sigmoid activation function after the last layer of the decoder. We used 20 Gaussian latent variables. The model was trained for 15K epochs using RMS optimizer with learning rate $1e-3$ and batch size 512.

Appendix B. Log likelihood estimates on the corrupted MNIST dataset

We evaluate the VAE V_1 and Jigsaw classifier Γ_1 described in Section 4 on binarised test images from the FashionMNIST dataset and corrupted binarised MNIST images. An example of an MNIST image corrupted with salt and pepper noise is shown in Figure 2(a). We evaluated the models using our proposed likelihood p_{FEVAE} and the PR evaluation scheme. The resulting log likelihood estimates are visualised in Figure 2(b). We observe the MNIST estimates fusing with the FashionMNIST estimates which demonstrates the influence of the sparse pixel distribution in binarised images as discussed in Section 4.



(a) Example of an original MNIST image (left) and the corrupted version (right).

(b) The PR log likelihoods on the FashionMNIST test split and corrupted MNIST test split under the p_{FEVAE} likelihood function.

Figure 2: Repeated experiment from Section 4 on the corrupted MNIST dataset.