2nd Symposium on Advances in Approximate Bayesian Inference, 2019 1–6

Arbitrarily-conditioned Data Imputation

Micael Carvalho[1,†], **Thibaut Durand**[2],
Jiawei He[1,†], **Nazanin Mehrasa**[1,†], and **Greg Mori**[1,2]

[1] *Simon Fraser University*
[2] *Borealis AI*

Abstract

In this paper, we propose an arbitrarily-conditioned data imputation framework built upon variational autoencoders and normalizing flows. The proposed model is capable of mapping any partial data to a multi-modal latent variational distribution. Sampling from such a distribution leads to stochastic imputation. Preliminary evaluation on MNIST dataset shows promising stochastic imputation conditioned on partial images as input.

1. Introduction

Neural network based algorithms have been shown effective and promising for various downstream tasks including classification (Deng et al., 2009; Damianou and Lawrence, 2013), retrieval (Carvalho et al., 2018), prediction (He et al., 2018), and more. In order to correctly learn how to perform these tasks, they usually rely strictly on access to fully-observed data. However, acquiring this type of data in real life requires tremendous human effort, limiting the applicability of this family of models. Having a framework designed to perform inference on partially-observed data will not only alleviate the aforementioned constraint, but also open possibilities to perform *data imputation*, in which the missing data is inferred.

Data imputation, also referred to conditional generation, has been an active research area (Little and Rubin, 1986; Song et al., 2018; Zadeh et al., 2019). The probabilistic nature of this task makes it difficult to adopt off-the-shelf deterministic models widely studied. In other words, conditioned on the same partially-observed data as input, multiple plausible fully-observed data should be able to be imputed. Variational autoencoders (VAEs) (Kingma and Welling, 2013), as a popular probabilistic modelling approach, have been applied to the data imputation task recently. A variational autoencoder defines a generative process that jointly models the distribution $p_\theta(\mathbf{x}, \mathbf{z})$ of the observed variable $\mathbf{x}$ and latent variable $\mathbf{z}$, governed by parameters θ. Instead of performing local inference, VAEs include an inference network parameterized by ϕ to output an approximate posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$. Both the generative model and the inference model are optimized with a unified evidence lower bound (ELBO) on marginal data likelihood: $\mathcal{L}_{\theta,\phi}(\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}}[q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]$. Recent literature on utilizing VAE-based models mainly focus on the effectiveness of combination of various obversed parts (Ma et al., 2019; Ivanov et al., 2018).

[†] Work developed during internship at Borealis AI

Different from the related works described above, we propose to enrich the latent space of variational autoencoders to enable multi-modal posterior inference, and therefore probabilistic imputation. Specifically, we use a two-stage model, with first-stage focusing on learning a representation space based on fully-observed data, and second-stage focusing on aligning the representation space embedded from partially-observed data to the one in stage-one. Using flow-based transformations for constructing a rich latent distribution, the proposed model is capable of inferring multi-modal variational latent distributions.

2. Imputation framework

Adopting a standard VAE approach for this problem would involve advocating for a model which receives partial data as input and, with the feedback of a standard reconstruction loss, learns to output the full data. Training such a model would pose many challenges. Firstly, gradient coming from the very end of the network would promote stronger imputation on the decoder, whereas the encoder could learn to simply encode the partial data. Secondly, there would be no mechanism to ensure the distribution of possible reconstructions would be correctly captured by the proposed posterior, which is generated by the encoder and fully conditioned on the partial data.

To amortize these problems, we propose a two-stage schema, represented in Figure 1. The first stage (upper part of the figure) corresponds to the encoder of a VAE model. This encoder was trained with an associated decoder, which was later discarded, with the task of encoding and reconstructing the full data. If properly trained, this stage's proposed posterior correctly depicts a good distribution of the full data, because this is a requirement in order to also reconstruct it. Once trained, its weights are fixed, and then the model of the second stage is trained on the partial data. Note that the encoders and decoders of the first and second stages are different – they can have the same architecture but do not share weights.

Because the latent space of the first model is rich enough to represent the full data's distribution (under the perspective of the first model), we propose to adopt a divergence loss between the first and the second model. This divergence acts as a distillation method, allowing the first model to inject rich information about the latent representation of the full data into the second-stage model. This injection will ensure weak alignment between both representation spaces, while also providing direct feedback to the encoder about the expected distribution of data in that space.

One problem with using simple families of posterior approximation is the lack of support for modeling multi-modal distributions, in which a reconstruction can take multiple forms. To compensate for that, we adopt a Normalizing Flow (Rezende and Mohamed, 2015) model inside the latent space, forcing the divergence between stage-one and stage-two to happen between the normal distribution, from the proposed posterior of the former, and the more complex distribution created by the flow model of the latter.

The nature of this divergence then becomes a problem: (1) How can we model a divergence between a simple and a more complex distribution for which we don't know the parameters? (2) Once defined, how can we ensure a multi-modal distribution *can* be modeled by the second stage?

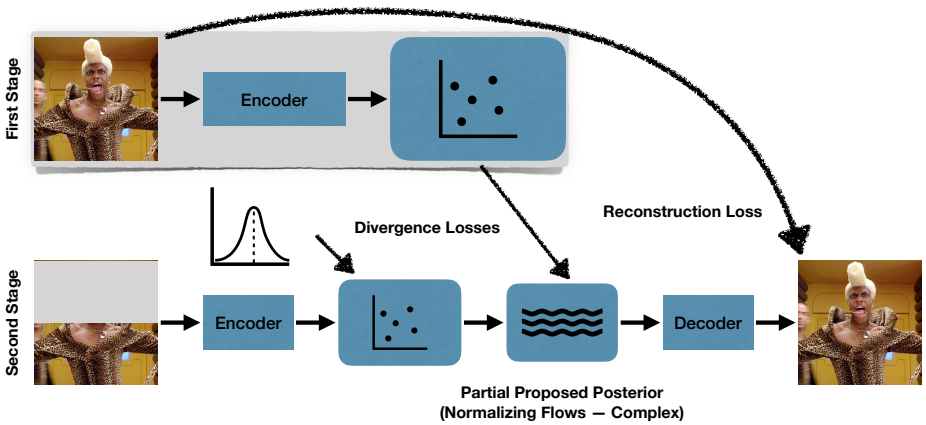


Figure 1: Representation of the proposed model. The first stage is trained observing the full data in the input, while the second stage is trained with partially observed data as input. The divergence loss between the proposed posteriors of the first and the second stages acts as a distillation loss, transferring knowledge to the second stage, and forcing the encoder to infer the distribution of the missing data.

To address (1), we examine relations between KL-Divergence and Likelihood Ratio (LR): $\text{LR} = \sum_{i=0}^{N} \frac{p(x_i)}{q(x_i)}$. From this perspective, we can derive a Monte-Carlo approach for the KL-Divergence, as long as $p(x_i)$ and $q(x_i)$ are tractable:

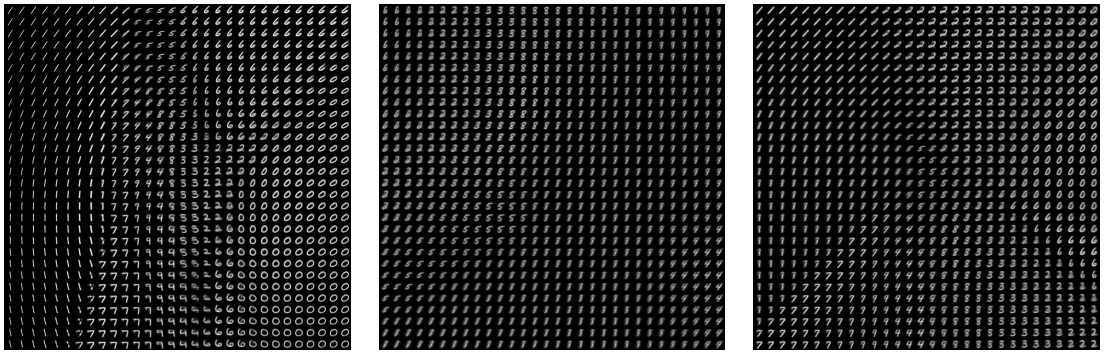
$$\text{KL}\left(p\left(\mathbf{x}\right)||q\left(\mathbf{x}\right)\right) := \mathbb{E}_{p(\mathbf{x})}\left[\ln\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right)\right] = \lim_{N\to\infty}\sum_{i=0}^{N}\ln\frac{p(x_i)}{q(x_i)}. \tag{1}$$

In our model, we know $p(x_i)$ is coming from a normal distribution, which is the proposed posterior of stage-one, therefore we only have to address the computation of $q(x_i)$, which is coming from the flow model. Thanks to properties of Normalizing Flows (NFs), this can be modeled as a correction term applied to the simple distribution before the flow:

$$\ln q_K(z_K) = \ln q_0(z_0) - \sum_{k=1}^{K}\ln\left|\det\frac{\partial f_k}{\partial z_{k-1}}\right|, \tag{2}$$

where K is the number of transformations f_k, and $q_0(z_0)$ is the simple distribution that is transformed to the complex distribution $q_K(z_K)$ through flow transformations.

To complete the model we also added a second divergence loss between the simple distribution (prior to the NF) and a Gaussian centered at zero with standard deviation of one. This extra divergence allows us to control the support of that distribution, regaining generative capability in all subsequent spaces, including the more complex one created by the NF module. The second stage model (encoder, partial posterior and decoder) is trained from scratch with the reconstruction and the divergence losses. During the training of the second stage model, the first stage model is fixed and it provides supervision for the structure of the latent space.



(*a*) Simple VAE, full data (*b*) 2(*a*) with partial data (*c*) 2(*b*) + normalizing flow

Figure 2: Samples following a regular grid in the 2-dimensional latent space generated by a simple VAE model which observes the full data 2(*a*), a VAE models with partial data 2(*b*), and a model with a normalizing flow module in the latent space 2(*c*).

(*a*) Task (*b*) Without Normalizing Flow (*c*) With Normalizing Flow

Figure 3: Reconstructions from partial data with and without normalizing flow.

Finally, problem (2) becomes irrelevant when we take into consideration the stochastic optimization in neural networks. If the training data is rich enough to correctly represent the multi-modal nature of the full data (this is a base assumption for any machine learning model), the best way to minimize the divergence loss is, indeed, by creating a multi-modal distribution which has density directly proportional to the likelihood of the data.

3. Analysis and Applications

In Figure 2 we present preliminary results which showcase the benefit of having each of the proposed modules. For this experiment, a regular grid is defined inside the latent space, and values in this grid are sampled from the decoder to observe the latent structure organization. In Figure 2(*a*), we display results for a baseline approach, representing the best possible scenario, in which the encoder has access to the full data. We then show, in Figure 2(*b*), the same space when adopting the schema in Figure 1, but without the NF module; and the full architecture – with NF – in Figure 2(*c*). We observe that the NF module allowed the network to have a more flexible latent space, when compared to the case without NF. Figure 2(*c*) displays clusters of digits which are clearly represented and well separated, unlike Figure 2(*b*), which exhibits classic problems of VAEs related to averaging images near cluster limits.

Following this experiment, we set out to test whether the multi-modality of reconstructions was being captured by the model. Due to limited space, we limit ourselves to a single example, for which we don't penalize the model for not perfectly reconstructing the partial data – the goal is to verify if the multi-modality is being captured, and if the model is

able to recognize the digit. Figure 3(a) demonstrates the problem we're aiming to solve: given a partially observed piece of data, we want to capture all possible interpretations and reconstructions of the full data. The results without NF and with NF are given in Figure 3(b) and Figure 3(c), respectively. We observe that adding the flow module allows the model to more precisely represent the possible reconstructions of partial data. While Figure 3(b) still displays signs of averaging and confusion, most of the digits in Figure 3(c) are clearly identifiable, and the multi-modality of the possible reconstructions is correctly depicted. Figure 3(b), for example, was unable to provide the possibility of "0" being a valid reconstruction to the partial provided in Figure 3(a).

Although we demonstrate the power of our model in the simple case of MNIST, our model remains data-agnostic, and can be applied to any data modality (images, videos, text, sound, etc). Possible applications range from arbitrarily-conditioned data imputation to data generation following complex modality interactions, which are partly modeled by the NF inside the latent space.

References

Micael Carvalho, Remi Cadene, David Picard, Laure Soulier, Nicolas Thome, and Matthieu Cord. Cross-modal retrieval in the cooking context: Learning semantic text-image embeddings. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '18, pages 35–44, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5657-2. doi: 10.1145/3209978.3210036.

Andreas Damianou and Neil Lawrence. Deep gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215, 2013.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Jiawei He, Andreas Lehrmann, Joseph Marino, Greg Mori, and Leonid Sigal. Probabilistic video generation using holistic attribute control. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–467, 2018.

Oleg Ivanov, Michael Figurnov, and Dmitry Vetrov. Variational autoencoder with arbitrary conditioning. 2018.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. URL http://arxiv.org/abs/1312.6114.

Roderick J A Little and Donald B Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, Inc., New York, NY, USA, 1986. ISBN 0-471-80254-9.

Chao Ma, Sebastian Tschiatschek, Konstantina Palla, Jose Miguel Hernandez-Lobato, Sebastian Nowozin, and Cheng Zhang. Eddi: Efficient dynamic discovery of high-value information with partial vae. In *International Conference on Machine Learning*, pages 4234–4243, 2019.

Danilo Jimenez Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. In *International Conference on Machine Learning (ICML)*, 2015.

Yuhang Song, Chao Yang, Zhe Lin, Xiaofeng Liu, Qin Huang, Hao Li, and C.-C. Jay Kuo. Contextual-based image inpainting: Infer, match, and translate. In *The European Conference on Computer Vision (ECCV)*, September 2018.

Amir Zadeh, Y. Lim, Paul Pu Liang, and Louis-Philippe Morency. Variational auto-decoder: Neural generative modeling from partial data. 2019.