ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

Conditional BRUNO: A Neural Process for Exchangeable Labelled Data

Iryna Korshunova[1], Yarin Gal[2], Arthur Gretton[3♠] and Joni Dambre[1♠] *

1- Ghent University- IDLab
Technologiepark-Zwijnaarde 15, 9052 Ghent - Belgium

2- University of Oxford - Computer Science Department
Oxford, OX1 3QD - UK

3- University College London - Gatsby Computational Neuroscience Unit
25 Howland Street, W1T 4JG London - UK

Abstract. We present a neural process that models exchangeable sequences of high-dimensional complex observations conditionally on a set of labels or tags. Our model combines the expressiveness of deep neural networks with the data-efficiency of Gaussian processes, resulting in a probabilistic model for which the posterior distribution is easy to evaluate and sample from, and the computational complexity scales linearly with the number of observations. The advantages of the proposed architecture are demonstrated on a challenging few-shot view reconstruction task which requires generalisation from short sequences of viewpoints.

1 Introduction

Exchangeability is an implicit assumption underlying many machine learning algorithms. It entails that any re-ordering of a finite sequence of observations is equally likely. As a consequence, it allows to reason about the future observations based on the behaviour of the previous ones. Owing to de Finetti's theorem, the exchangeability property is a cornerstone of Bayesian statistics as it facilitates inference and parameter learning in probabilistic models.

Some problems can be explicitly formulated in terms of modelling exchangeable data. For instance, few-shot concept learning can be seen as learning to complete short exchangeable sequences [1], where it is natural to assume no inherent ordering in the observations coming from the same concept. BRUNO [2] follows the explicit approach by modelling autoregressive distributions $p(x_n|x_{1:n-1})$ of an exchangeable process. This was proven to be an efficient way of doing both few-shot image generation and classification within one model.

In this work, we extend the idea of BRUNO to the conditional case, where we wish to model $p(x_n|h_n, x_{1:n-1}, h_{1:n-1})$, where h_i are labels or tags associated with observations x_i. One example of where conditional BRUNO can be used, is a task of generating new viewpoints of an object or a scene while given a few images of that scene under different camera positions.

*We would like to thank Jonas Degrave and Ferenc Huszár for their conceptual contributions to this work. ♠ Shared authorship.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

Formally, a stochastic process $x_1, x_2, x_3, \ldots$ is said to be exchangeable if for all n and all permutations π

$$p(x_1, \ldots, x_n) = p\left(x_{\pi(1)}, \ldots, x_{\pi(n)}\right), \tag{1}$$

i.e. the joint probability remains the same under any permutation of the sequence.

The concept of exchangeability is intimately related to Bayesian statistics via de Finetti's theorem, which states that every exchangeable process is a mixture of i. i. d. processes:

$$p(x_1, \ldots, x_n) = \int p(\theta) \prod_{i=1}^{n} p(x_i|\theta) d\theta, \tag{2}$$

where θ is a parameter vector (finite or infinite-dimensional) conditioned on which, the x_i's are i. i. d. [3].

This theorem gives two ways of defining models of exchangeable sequences. One is via explicit Bayesian modelling: define a prior $p(\theta)$, a likelihood $p(x_i|\theta)$ and calculate the posterior in Eq. 2 directly. Here, the difficulty is the intractability of the posterior as it requires an integration over the parameter θ. A common solution is to use a variational approximation. The neural statistician [4] implements this approach by building upon a variational autoencoder model (VAE) [5].

The second way is to *construct* an exchangeable process while modelling its autoregressive distributions $p(x_n|x_{1:n-1})$ directly without referring to the underlying Bayesian model. BRUNO [2] proposes a design for doing so. It consists of two components: **(a)** a bijective mapping that transforms an intricate input space $\mathcal{X}$ into a Gaussian latent space $\mathcal{Z}$, and **(b)** a collection of exchangeable Gaussian processes ($\mathcal{GP}$s) defined in the latent space $\mathcal{Z}$. Using deep neural networks to implement the bijection $f : \mathcal{X} \mapsto \mathcal{Z}$ allows to model complex and high-dimensional inputs. At the same time, the construction of BRUNO guarantees that the process in $\mathcal{X}$ is exchangeable, and thus the model performs an exact, albeit implicit, Bayesian inference in space $\mathcal{X}$.

A natural extension when building exchangeable models would be to have a conditional process with two associated sequences: $x_1, x_2, x_3, \ldots$ and $h_1, h_2, h_3, \ldots$. For instance, when $\boldsymbol{x}_i$ is an image and $\boldsymbol{h}_i$ a vector of descriptive labels or tags. By analogy with Eq. 1, the exchangeability property becomes:

$$p(x_1, \ldots, x_n|h_1, \ldots, h_n) = p\left(x_{\pi(1)}, \ldots, x_{\pi(n)}|h_{\pi(1)}, \ldots, h_{\pi(n)}\right). \tag{3}$$

To have a valid stochastic process, we also need a consistency property as imposed by the Kolmogorov extension theorem [6]:

$$p(x_{1:m}|h_{1:m}) = \int p\left(x_{1:n}|h_{1:n}\right) dx_{m+1:n} \text{ for } 1 \leq m < n. \tag{4}$$

To our best knowledge, Bayesian theory does not have an established proof of de Finetti's theorem for conditional probabilities. Namely, that the two conditions above ensure that one can represent the process as a mixture of

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

conditionally i.i.d. models as given in Eq. 5. For the processes where x_i and h_i take values from a finite set, this theorem is proven in the field of quantum physics [7]. However, it is yet unclear how to extend their results to continuum variables.

$$p(x_{1:n}|h_{1:n}) = \int p(\theta) \prod_{i=1}^{n} p(x_i|h_i, \theta) d\theta \text{ or equivalently,} \tag{5}$$

$$p(x_n|h_n, x_{1:n-1}, h_{1:n-1}) = \int p(\theta|x_{1:n-1}, h_{1:n-1}) p(x_n|h_n, \theta) d\theta \tag{6}$$

Relying on the conditional version of de Finetti's theorem, neural processes [8] take an approach that is similar to the neural statistician's. It extends the VAE model to handle collections of (x_i, h_i) input pairs and dealing with a variational lower bound on $p(x_n|h_n, x_{1:n-1}, h_{1:n-1})$. Versa [9] also follows the idea of approximating the aforementioned posterior predictive distribution, though it uses a training procedure that differs from the standard variational inference. Both models achieve permutation invariance of $p(\theta|x_{1:n}, h_{1:n})$ with respect to the conditioning inputs by using instance-pooling operations, e.g. the mean over representations of (x_i, h_i) pairs.

Another option that does not require approximation of the right-hand side of Eq. 6, is to use the idea of BRUNO and construct a process that satisfies Eq. 3 and Eq. 4 directly. In the next section, we show this can be done by slightly modifying the architecture of BRUNO. Namely, by conditioning the bijective transformation $f : \mathcal{X} \mapsto \mathcal{Z}$ on the tags, such that $\boldsymbol{z}_i = f_{\boldsymbol{h}_i}(\boldsymbol{x}_i)$. A schematic of our model is given in Fig. 1.

2 Conditional BRUNO

The bijective transformation part of BRUNO is carried out by a Real NVP [10] – a deep, invertible and learnable neural network architecture that transforms some density $p(\boldsymbol{x})$ into a desired probability distribution $p(\boldsymbol{z})$. It is implemented as a sequence of alternating coupling layers, with every layer transforming a half of its input dimensions while copying the other half directly to the output. In case of modelling a conditional distribution $p(\boldsymbol{x}|\boldsymbol{h})$, we can make the transformation dependendent on $\boldsymbol{h}$, so the outputs of the coupling layer become:

$$\begin{cases} \boldsymbol{x}_{\text{out}}^{1:d} = \boldsymbol{x}_{\text{in}}^{1:d} \\ \boldsymbol{x}_{\text{out}}^{d+1:D} = \boldsymbol{x}_{\text{in}}^{d+1:D} \odot \exp(s(\boldsymbol{x}_{\text{in}}^{1:d}, \boldsymbol{h})) + t(\boldsymbol{x}_{\text{in}}^{1:d}, \boldsymbol{h}), \end{cases} \tag{7}$$

where $\odot$ is an elementwise product, and functions s (scale) and t (translation) are usually deep neural networks. We achieved the conditioning on $\boldsymbol{h}$ by concatenating the features of $\boldsymbol{h}$ to the inputs of every dense and convolutional layer inside s and t networks.

As in case of the original Real NVP model, we can assume a fixed distribution for the latents $\boldsymbol{z}$ due to the fact that dependence of $\boldsymbol{x}$ on $\boldsymbol{h}$ is introduced

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

via the Jacobian of the transformation. The latter is used in the change of variables formula: $p(\boldsymbol{x}|\boldsymbol{h}) = p(\boldsymbol{z})\,|\det \boldsymbol{J_h}|$. For the same reasons, the conditional BRUNO can use the two assumptions below, which are identical to those from its unconditional counterpart.
A1: dimensions $\{z^d\}_{d=1,\dots,D}$ are independent, so $p(\boldsymbol{z}) = \prod_{d=1}^{D} p(z^d)$
A2: for each dimension d, we assume that $(z_1^d, \dots z_n^d) \sim MVN_n(\mathbf{0}, \boldsymbol{\Sigma}^d)$, where $\boldsymbol{\Sigma}^d$ is a $n \times n$ covariance matrix with $\boldsymbol{\Sigma}^d_{ii} = v^d$ and $\boldsymbol{\Sigma}^d_{ij,i\neq j} = \rho^d$, $0 \leq \rho^d < v^d$.

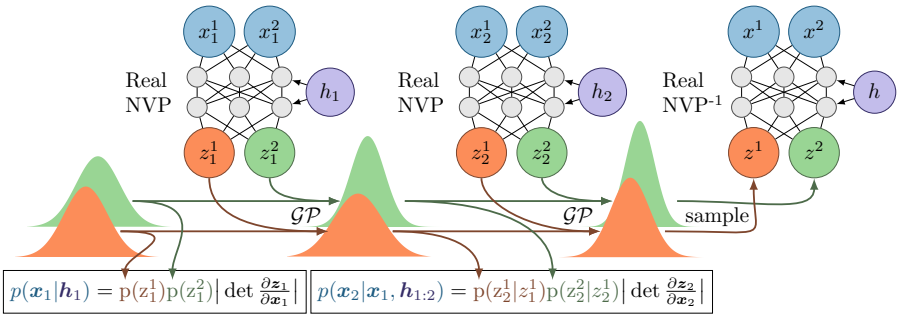

Fig. 1: A schematic of the conditional BRUNO model.

3 Experiments

We consider a task of few-shot image reconstruction, where the model is required to infer how an object looks from various angles based on a small set of observed views [9]. This problem can be framed as generating samples from a predictive conditional distribution $p(\boldsymbol{x}_n|\boldsymbol{h}_n, \boldsymbol{x}_{1:n-1}, \boldsymbol{h}_{1:n-1})$, where $\boldsymbol{h}_n$ is a desired angle and $\boldsymbol{x}_{1:n-1}$ is a set of observed views associated with angles $\boldsymbol{h}_{1:n-1}$. We use airplanes and chairs from the ShapeNetCore v2 [11] dataset as constructed in [9], and train the conditional BRUNO on one-shot tasks. Namely, we give a single random view $\boldsymbol{x}_1$ and its angle $\boldsymbol{h}_1$ and the goal is to predict N views of the same object under angles $\boldsymbol{h}_1, \dots, \boldsymbol{h}_N$. In this case, the objective is to maximise $\mathcal{L} = \sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\boldsymbol{h}_n, \boldsymbol{x}_1, \boldsymbol{h}_1)$ with respect to the Real NVP parameters and variance-covariance parameters of the latent $\mathcal{GP}$s. Note that unlike in [9], we train a single model on a combined set of chairs and airplanes. The code to reproduce our experiments is available at `github.com/IraKorshunova/bruno`.

In Figure 2, we show samples from a conditional BRUNO when the model was given a single viewpoint from an object not seen during training. Note that conditioned on a chair, it never samples an airplane or vice versa. Moreover, one can see how the model's uncertainty about the object is reflected in the samples. Specifically, when the single shot it is conditioned upon gives insufficient information about the object, conditional BRUNO generates diverse objects which are consistent with the given shot. Also, our samples always have a correct orientation and their quality is comparable to the one from Versa [9], though, our samples are sharper.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

A more difficult few-shot image reconstruction task is learning to render scenes as done by Generative Query Networks(GQN) [12], which are similar to the neural processes [8] in their core idea. We hypothesize that conditional BRUNO, when scaled to more complex datasets, might become a viable and simpler alternative to the GQN-style types of models. Though, we could not conduct those experiments due to huge computational demands of the models.

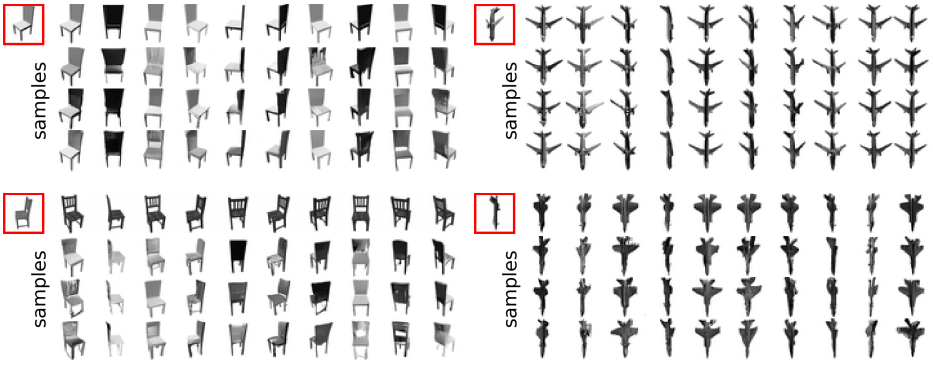

Fig. 2: One-shot BRUNO samples for the unseen test objects. Here, we condition on a single view $(\boldsymbol{x}_1, \boldsymbol{h}_1)$ of a chair or an airplane. The input shot is marked in red. On the top row is the ground truth, whereas the three rows underneath contain our samples from $p(\boldsymbol{x}|\boldsymbol{h}, \boldsymbol{x}_1, \boldsymbol{h}_1)$ conditioned on the input shot and a desired angle $\boldsymbol{h}$. In the bottom two cases, it is difficult to infer the exact appearance from the single shot. For the chair, it is not possible to deduce the back style, and for the airplane the shape of the wings. Consequently, the model increases the variability of the samples along these dimensions.

4 Discussion and conclusion

We showed that BRUNO [2] can be easily extended to the conditional case while maintaining its appealing properties such as **(a)** exact likelihoods **(b)** fast sampling and inference, **(c)** no retraining or changes to the architecture at test time, and **(d)** recurrent formulation. These features make a simple yet an effective and flexible model for meta-learning. Together, conditional and unconditional BRUNO cover a broad range of meta-learning tasks from a few-shot conditional image generation to online set anomaly detection.

BRUNO builds directly on the fundamental property of exchangeability that underlies much of Bayesian statistics. Therefore, it provides an alternative, previously unexplored way for building meta-learning models. Firstly, it abandons the approximate explicit Bayesian inference in favour of an exact and implicit one. The latter property, however, does not seem to limit the applicability of BRUNO compared to other meta-learning methods. Secondly, it probes the exchangeable $\mathcal{GP}s$ instead of a prevalent mean aggregator for integrating the information about

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

all inputs in a permutation-invariant way. It remains unclear, however, which of the two approaches works best in practice and what their failure cases are.

BRUNO combines $\mathcal{GP}$s with powerful bijective feature extractors in the form of flow-based deep neural architectures, and while the former is unlikely to be improved, we expect BRUNO to greatly benefit from the recent advances in building normalising flows, which is currently an active area of research [13, 14].

References

[1] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 2015.

[2] I. Korshunova, J. Degrave, F. Huszár, Y. Gal, A. Gretton, and J. Dambre. BRUNO: A deep recurrent model for exchangeable data. In *Proceedings of the 32th International Conference on Neural Information Processing Systems*, 2018.

[3] D. J. Aldous, P. L. Hennequin, I. A. Ibragimov, and J. Jacod. *Ecole d'Ete de Probabilites de Saint-Flour XIII, 1983*. Lecture Notes in Mathematics. Springer Berlin Heidelberg, 1985.

[4] H. Edwards and A. Storkey. Towards a neural statistician. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[5] D. P Kingma and M. Welling. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations*, 2014.

[6] B. Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Hochschultext / Universitext. Springer, 2003.

[7] J. Barrett and M. Leifer. The de Finetti theorem for test spaces. *New Journal of Physics*, 11(3), 2009.

[8] M. Garnelo, J. Schwarz, D. Rosenbaum, F. Viola, D. J. Rezende, S. M. A. Eslami, and Yee Whye Teh. Neural processes. *Theoretical Foundations and Applications of Deep Generative Models, ICML workshop*, 2018.

[9] J. Gordon, J. Bronskill, M. Bauer, S. Nowozin, and R. E Turner. Decision-theoretic meta-learning: Versatile and efficient amortization of few-shot learning. *arXiv preprint arXiv:1805.09921*, 2018.

[10] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using Real NVP. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[11] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q.-X. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, Li Yi, and F. Yu. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[12] S. M. A. Eslami, D. Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, D. P. Reichert, L. Buesing, T. Weber, O. Vinyals, D. Rosenbaum, N. Rabinowitz, H. King, C. Hillier, M. Botvinick, D. Wierstra, K. Kavukcuoglu, and D. Hassabis. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.

[13] W. Grathwohl, R. T. Q. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

[14] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018.