Pattern Recognition 107 (2020) 107501

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Handling incomplete heterogeneous data using VAEs

Alfredo Nazábal[a,*], Pablo M. Olmos[b], Zoubin Ghahramani[c,d], Isabel Valera[e,f]

[a] The Alan Turing Institute, London, United Kingdom
[b] University Carlos III, Madrid, Spain
[c] University of Cambridge, Cambridge, United Kingdom
[d] Uber AI Labs, San Francisco, US
[e] Max Planck Institute for Intelligent Systems, Tübingen, Germany
[f] Department of Computer Science, Saarland University, Saarbrücken, Germany

ARTICLE INFO

Article history:
Received 25 March 2019
Revised 23 March 2020
Accepted 12 June 2020
Available online 13 June 2020

Keywords:
Generative models
Variational autoencoders
Incomplete heterogenous data

ABSTRACT

Variational autoencoders (VAEs), as well as other generative models, have been shown to be efficient and accurate for capturing the latent structure of vast amounts of complex high-dimensional data. However, existing VAEs can still not directly handle data that are heterogenous (mixed continuous and discrete) or incomplete (with missing data at random), which is indeed common in real-world applications.

In this paper, we propose a general framework to design VAEs suitable for fitting incomplete heterogenous data. The proposed HI-VAE includes likelihood models for real-valued, positive real valued, interval, categorical, ordinal and count data, and allows accurate estimation (and potentially imputation) of missing data. Furthermore, HI-VAE presents competitive predictive performance in supervised tasks, outperforming supervised models when trained on incomplete data.

Data are usually organized and stored in databases, which are often large, heterogenous, noisy, and incomplete. For example, an online shopping platform has access to heterogenous and incomplete information of its users, such as their age, gender, orders, wish lists, etc. Similarly, Electronic Health Records of hospitals might contain different lab measurements, diagnoses and genomic information about their patients. Learning generative models that accurately capture the distribution, and therefore the underlying latent structure, of such incomplete and heterogeneous datasets may allow us to better understand the data, estimate missing or corrupted values, detect outliers, and make predictions (e.g., on patients' diagnosis) on unseen data [1].

Deep generative models have been recently proved to be highly flexible and expressive unsupervised methods, able to capture the latent structure of complex high-dimensional data. They efficiently emulate complex distributions from large high-dimensional datasets, generating new data points similar to the original real-world data, after training is completed [2–4]. So far, the main focus in the literature is to enrich the prior or posterior of explicit generative models such as variational autoencoders (VAEs); or to propose alternative training objectives to the log-likelihood, leading to implicit generative models such as, e.g., generative adversarial networks (GANs) [5]. Indeed, we are witnessing a race between an ever-growing spectrum of VAE models, e.g., VAE with a VampPrior [6], Output Interpretable VAEs [7], DVAE++ [8], Shape Variational Autoencoder [9] and GAN-style objective functions (f-GAN [10], DR-GAN [11], Wasserstein GANs [12], MMD-GAN [13], Gated-GAN [14], AdaGAN [15], feature-matching GAN [16], etc.). While all these approaches compete to generate the most realistic images or readable text, the deployment of such models to solve practically-relevant problems in arbitrary datasets, which are often incomplete and heterogenous [17], is being overlooked. In the following, we discuss these problems in more detail and why we believe our paper is relevant to data-scientists interested in exploiting the deep generative model pipeline in the data wrangling process. We provide with practical tools to handle both missing and heterogeneous data with little supervision from the user, who merely has to indicate the data type model of each attribute and the position of the missing data.

Currently deep generative models focus on highly-structured homogeneous data collections including, e.g., images [18,19], text [20], video [21–23] or speech [24], which are characterized by strong statistical dependencies between pixels or words. The dominant existing approach to account for heterogenous data follows a deep domain-alignment approach [25–27], designed to discover

[*] Corresponding author.
E-mail addresses: anazabal@turing.ac.uk (A. Nazábal), olmos@tsc.uc3m.es (P.M. Olmos), zoubin@eng.cam.ac.uk (Z. Ghahramani), ivalera@cs.uni-saarland.de (I. Valera).

https://doi.org/10.1016/j.patcog.2020.107501
0031-3203/© 2020 Elsevier Ltd. All rights reserved.

relations between two unpaired unlabelled datasets rather than modelling their joint distribution using a probabilistic generative model [28–30]. Surprisingly, not much attention has been paid to describing how deep generative models can be designed to effectively learn the distribution of less structured, heterogeneous datasets. In these datasets there is no clear notion of correlation among the different attributes (or dimensions) to be exploited by weight sharing using convolutional or recurrent neural networks. As we show in this paper, preventing a few dimensions of the data dominating the training is a crucial aspect to effectively deploy deep generative models suitable for heterogeneous data.

Similarly, there is no clear discussion in the current literature on how to incorporate missing data during the training of deep generative models. Existing approaches consider either complete data during training [31], or assume incomplete information only in one of the dimensions of the data, which corresponds to the one they aim to predict (e.g., the label in a classification task) [32,33]. However, both approaches are not realistic enough, since it might be crucial for the performance of an unsupervised model to use all the available information during training. Recently, [34] proposed a GAN approach, named as GAIN, to input missing data, where the generator completes the missing values given the observed ones, and the discriminator aims to distinguish between true and imputed values. However, this approach can only handle continuous or binary data, and it is not easily generalizable to heterogeneous data. As a consequence, strategies to effectively train deep generative models on incomplete and heterogeneous datasets are still required.

In this work, we present a general framework for VAEs that effectively incorporates incomplete data and heterogenous observations. Our design presents the following features:

i) a generative model that handles mixed numerical (continuous real-valued and positive real-valued, as well as discrete count data) and nominal (categorical and ordinal data) likelihood models, which we parametrize using deep neural networks (DNNs);
ii) a stable recognition model that handles Missing Data Completely at Random (MCAR) without increasing its complexity or promoting overfitting;
iii) a data-normalization input/output layer that prevents a few dimensions of the data dominating the training of the VAE, improving also the training convergence; and
iv) an ELBO (Evidence Lower Bound), used to optimize the parameters of both the generative and the recognition models, that is computed only on the observed data, regardless of the pattern of missing data.

The resulting VAE is a fully unsupervised model which allows us not only to accurately solve unsupervised tasks, such as density estimation or missing data completion, but also supervised tasks (e.g., classification or regression) with incomplete input data. We provide the reader with specific guidelines to design VAEs for real-world data, which are compatible with modern efforts in the design of VAEs and implicit models (GANs), mainly oriented to prevent the mode-dropping effect [12,35]. Our empirical results show that our proposal outperforms competitors, including the recent GAIN [34], on a heterogenous data completion task, and provides comparable accuracy in classification tasks to deep supervised methods–which cannot handle missing values in the input data, therefore, requiring imputing missing inputs in the data.

1. Problem statement

We define a heterogeneous dataset as a collection of N objects, where each object is defined by D attributes and these attributes correspond to either numerical (continuous or discrete) or nominal variables. We denote each object in the dataset as a D-dimensional vector $\mathbf{x}_n = [x_{n1}, \ldots, x_{nD}]$, where each attribute x_{nd} corresponds to one of the following data types:

- Numerical variables:
  1. Real-valued data, which takes values in the real line, i.e., $x_{nd} \in \mathbb{R}$.
  2. Positive real-valued data, which takes values in the positive real line, i.e., $x_{nd} \in \mathbb{R}^+$.
  3. (Discrete) count data, which takes values in the natural numbers, i.e., $x_{nd} \in \{1, \ldots, \infty\}$.
- Nominal variables:
  1. Categorical data, which takes values in a finite unordered set, e.g., $x_{nd} \in$ {‘blue’, ‘red’, ‘black’}.
  2. Ordinal data, which takes values in a finite ordered set, e.g., $x_{nd} \in$ {‘never’, ‘sometimes’, ‘often’, ‘usually’, ‘always’}.

Additionally, we consider that a random set of entries in the data is incomplete, under the MCAR assumption [36], such that each object $\mathbf{x}_n$ can potentially correspond to any combination of observed and missing attributes. Let $\mathcal{O}_n$ ($\mathcal{M}_n$) be the index set of observed (missing) attributes for the nth data point, where $\mathcal{O}_n \cap \mathcal{M}_n = \emptyset$. Also, let $\mathbf{x}_n^o$ ($\mathbf{x}_n^m$) represent the sliced $\mathbf{x}$ vector, including only the elements indexed by $\mathcal{O}_n$ ($\mathcal{M}_n$). Fig. 1(a) shows an example of an incomplete heterogenous dataset, where we observe that the different attributes (or dimensions) in the data correspond to different types of numerical and nominal variables, and missing values appear ubiquitously across the data.

Diverging from common trends in the deep generative community, we consider databases that do not contain highly-structured homogeneous data, but instead each observed object is a set of scalar mixed numerical and nominal attributes, being the correlations between attributes (the underlying structure), in many cases, weak. Since the dimensionality of these datasets can be relatively small (compared to images for instance), we need to carefully design the generative model to avoid overfitting on the observed data, while keeping the model flexible enough to incorporate both heterogeneous data types and random patterns of missing data.

2. Generalizing VAEs for heterogeneous and incomplete data

In this section, we show how to extend the vanilla VAE introduced in [2] to handle incomplete and heterogeneous data.

2.1. Handling incomplete data

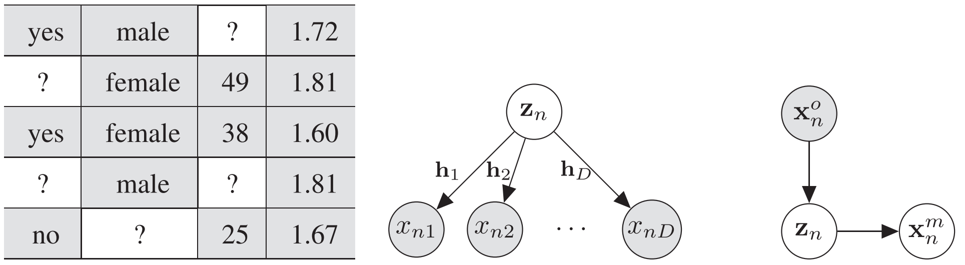
In a standard VAE, missing data affect both the generative (decoder) and the recognition (encoder) models. The ELBO is defined over the complete data, and it is not straightforward to decouple the missing entries from rest of the data, particularly when these entries appear completely at random in the dataset. To this end, we first propose to use the following factorization for the decoder (Fig. 1(b)):

$$p(\mathbf{x}_n, \mathbf{z}_n) = p(\mathbf{z}_n) \prod_d p(x_{nd}|\mathbf{z}_n), \tag{1}$$

where $\mathbf{z}_n \in \mathbb{R}^K$ is the latent K-dimensional vector representation of the object $\mathbf{x}_n$, and $p(\mathbf{z}_n) = \mathcal{N}(\mathbf{z}_n|\mathbf{0}, \mathbf{I}_K)$. This factorization makes it easy to marginalize out the missing attributes for each object from the variational ELBO. We parametrize the likelihood $p(x_{nd}|\mathbf{z}_n)$ with the set of parameters $\boldsymbol{\gamma}_{nd} = \mathbf{h}_d(\mathbf{z}_n)$, where $\mathbf{h}_d(\mathbf{z}_n)$ is a DNN that transforms the latent variable $\mathbf{z}_n$ into the likelihood parameters $\boldsymbol{\gamma}_{nd}$.

Note that the above factorization of the likelihood allows us to separate the contributions of the observed data $\mathbf{x}_n^o$ from the missing data $\mathbf{x}_n^m$ as

$$p(\mathbf{x}_n|\mathbf{z}_n) = \prod_{d \in \mathcal{O}_n} p(x_{nd}|\mathbf{z}_n) \prod_{d \in \mathcal{M}_n} p(x_{nd}|\mathbf{z}_n). \tag{2}$$



yes	male	?	1.72
?	female	49	1.81
yes	female	38	1.60
?	male	?	1.81
no	?	25	1.67

(a) Database example (b) Generative Model (c) Recognition Model

Fig. 1. (a) Example of incomplete heterogenous data. Panel (b) shows our generative model, where every dimension in the observation vector $\mathbf{x}_n = [x_{n1}, \ldots, x_{nD}]$ corresponds to either a numerical or nominal variable, and therefore, the likelihood parameters of each dimension d are independently provided by an independent DNN $\mathbf{h}_d$. Additionally, panel (c) shows our recognition model to infer the missing data $\mathbf{x}_n^m$ from observed data $\mathbf{x}_n^o$.

The recognition model, graphically represented in Fig. 1(c), also needs to account for incomplete data, such that the distribution of the latent variable $\mathbf{z}_n$ only depends on the observed attributes $\mathbf{x}_n^o$, i.e.,

$$q(\mathbf{z}_n, \mathbf{x}_n^m|\mathbf{x}_n^o) = q(\mathbf{z}_n|\mathbf{x}_n^o) \prod_{d\in\mathcal{M}_n} p(x_{nd}|\mathbf{z}_n). \tag{3}$$

Given the above generative and recognition models, described respectively by (1) and (3), the ELBO of the marginal likelihood (computed only on the observed data $\mathbf{X}^o$) can be written as

$$\begin{aligned}\log p(\mathbf{X}^o) &= \sum_{n=1}^{N} \log p(\mathbf{x}_n^o) = \sum_{n=1}^{N} \log \int p(\mathbf{x}_n^o, \mathbf{x}_n^m, \mathbf{z}_n) d\mathbf{z}_n \, d\mathbf{x}_n^m \\ &\geq \sum_{n=1}^{N} \mathbb{E}_{q(\mathbf{z}_n|\mathbf{x}_n^o)}\left[\sum_{d\in\mathcal{O}_n} \log p(x_{nd}|\mathbf{z}_n)\right] \\ &\quad - \sum_{n=1}^{N} \mathrm{KL}(q(\mathbf{z}_n|\mathbf{x}_n^o)||p(\mathbf{z}_n)),\end{aligned} \tag{4}$$

where the first term of the ELBO corresponds to the reconstruction term of (*only*) the observed data $\mathbf{X}^o$, and the Kullback-Liebler (KL) divergence in the second term penalizes any deviation of the posterior $q(\mathbf{z}_n|\mathbf{x}_n^o)$ from the prior $p(\mathbf{z}_n)$. Note that the KL divergence can be computed in closed-form [2].

Recognition models for incomplete data.

We need an encoder that is flexible enough to handle any combination of observed and missing attributes. To this end, we propose an *input drop-out* recognition distribution whose parameters are the output of a DNN with input $\tilde{\mathbf{x}}_n$, such that

$$q(\mathbf{z}_n|\mathbf{x}_n^o) = \mathcal{N}\big(\mathbf{z}_n|\boldsymbol{\mu}_q(\tilde{\mathbf{x}}_n), \boldsymbol{\Sigma}_q(\tilde{\mathbf{x}}_n)\big), \tag{5}$$

where the input $\tilde{\mathbf{x}}_n$ is a D-length vector that resembles the original observed vector $\mathbf{x}_n$ but the missing dimensions are replaced by zeros, and $\boldsymbol{\mu}_q(\tilde{\mathbf{x}}_n)$ and $\boldsymbol{\Sigma}_q(\tilde{\mathbf{x}}_n)$ are parametrized DNNs with input $\tilde{\mathbf{x}}_n$ whose output determine the mean and the diagonal covariance matrix of (5). In order to make sure that the missing inputs do not affect to the output of the encoder (nor to the learning of its parameters), we need to ensure that the contribution of the missing attributes to the encoder outputs and the evaluation of the derivatives with respect to the network parameters of $\boldsymbol{\mu}_q(\tilde{\mathbf{x}}_n)$ and $\boldsymbol{\Sigma}_q(\tilde{\mathbf{x}}_n)$ is zero. To this end, we rely on multilayer perceptron neural network architectures, where the output of every neuron is a non-linear transformation of a (linear) weighted sum of the inputs, and thus the output (and its derivative) does not depend on the zero entries.

An alternative approach, proposed in [37], consists of exploiting the properties of Gaussian distributions in the linear factor analysis case [38] and extending them to non-linear models, designing a factorized recognition model:

$$q(\mathbf{z}_n|\mathbf{x}_n^o) = p(\mathbf{z}_n) \prod_{d\in\mathcal{O}_n} q(\mathbf{z}_n|x_{nd}),$$

where $q(\mathbf{z}_n|x_{nd}) = \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}_d(x_{nd}), \boldsymbol{\Sigma}_d(x_{nd}))$, and therefore, $q(\mathbf{z}_n|\mathbf{x}_n^o) = \mathcal{N}\big(\mathbf{z}_n|\boldsymbol{\mu}_q(\mathbf{x}_n^o), \boldsymbol{\Sigma}_q(\mathbf{x}_n^o)\big)$ with

$$\boldsymbol{\Sigma}_q^{-1}(\mathbf{x}_n^o) = \mathbf{I}_K + \sum_{d\in\mathcal{O}_n} \boldsymbol{\Sigma}_d^{-1}(x_{nd}), \tag{6}$$

$$\boldsymbol{\mu}_q(\mathbf{x}_n^o) = \boldsymbol{\Sigma}_q(\mathbf{x}_n^o)\left(\sum_{d\in\mathcal{O}_n} \boldsymbol{\mu}_d(x_{nd})\boldsymbol{\Sigma}_d^{-1}(x_{nd})\right). \tag{7}$$

Note that, in contrast to our input drop-out recognition model, in this case we need to train an independent DNN per attribute d, which might not only result in a higher computational cost, as well as in overfitting, but it also loses the ability of DNNs to amortize the inference of the parameters across attributes, and therefore, across different missing data patterns.

Remark. This VAE for incomplete data can readily be used to estimate the missing values in the data as follows

$$p(\mathbf{x}_n^m|\mathbf{x}_n^o) \approx \int p(\mathbf{x}_n^m|\mathbf{z}_n) q(\mathbf{z}_n|\mathbf{x}_n^o) d\mathbf{z}_n \tag{8}$$

The KL term in (4), promotes a missing-data recognition model that does not rely on the observed attributes, i.e., $p(\mathbf{x}_n^m|\mathbf{x}_n^o) \approx \int p(\mathbf{x}_n^m|\mathbf{z}_n)\mathcal{N}(\mathbf{z}_n|\mathbf{0}, \mathbf{I}_K)d\mathbf{z}_n$. In those cases where the KL term in (4) tends to dominate the ELBO, we can modify the probabilistic model to favour richer structures in the posterior distribution by replacing the independent Gaussian prior with a more structured distribution such as a mixture model. We discuss this approach in more detail in Section 3.

2.2. Handling heterogenous data

Standard applications of VAEs consider homogeneous data (e.g., images) where all the observed attributes (pixels) share the likelihood function (e.g., a Gaussian or a Bernoulli distribution [2]) whose parameters are often jointly modeled by a single NN (e.g., a convolutional DNN). In contrast, our setting assumes that every attribute in the data may correspond to one of the numerical or nominal data types introduced in Section 1, and thus it requires an appropriate likelihood function. Assuming the factorized decoder in (1), we can easily accommodate a variety of likelihood functions, one per input attribute, where an independent DNN, $\mathbf{h}_d(\cdot)$, is used to determine the parameters $\boldsymbol{\gamma}_{nd}$ of every likelihood model $p(x_{nd}|\mathbf{z}_n) = p(x_{nd}|\boldsymbol{\gamma}_{nd} = \mathbf{h}_d(\mathbf{z}_n))$, as shown in Fig. 1(b). Next, we define suitable likelihood functions to model the numerical and nominal data types introduced in Section 1, and show how to parameterize these likelihood functions using DNNs. We remark, that while here we have selected common choice likelihood functions as showcase examples – e.g., log-Normal, ordinal logit-function and Poisson distributions to respectively model positive real-valued, ordinal categorical nominal, and count variables –, other distributions, such as the Gamma distribution for positive real data or the negative binomial distribution for count data, could alternatively be used.

1. Real-valued data. For real-valued data, we assume a Gaussian likelihood model, i.e.,

$$p(x_{nd}|\boldsymbol{\gamma}_{nd}) = \mathcal{N}\big(x_{nd}|\mu_d(\mathbf{z}_n), \sigma_d^2(\mathbf{z}_n)\big), \tag{9}$$

with $\boldsymbol{\gamma}_{nd} = \{\mu_d(\mathbf{z}_n), \sigma_d^2(\mathbf{z}_n)\}$, where the mean $\mu_d(\mathbf{z}_n)$ and the variance $\sigma_d^2(\mathbf{z}_n)$ are computed as the outputs of DNNs with input $\mathbf{z}_n$.

2. Positive real-valued data. For positive real-valued data, we assume a log-normal likelihood model, i.e.,

$$p(x_{nd}|\boldsymbol{\gamma}_{nd}) = \log\mathcal{N}\big(x_{nd}|\mu_d(\mathbf{z}_n), \sigma_d^2(\mathbf{z}_n)\big), \tag{10}$$

with $\boldsymbol{\gamma}_{nd} = \{\mu_d(\mathbf{z}_n), \sigma_d^2(\mathbf{z}_n)\}$, where the likelihood parameters $\mu_d(\mathbf{z}_n)$ and $\sigma_d^2(\mathbf{z}_n)$ (which corresponds to the mean and variance of the variable's natural logarithm) are the outputs of DNNs with input $\mathbf{z}_n$.

3. Count data. For count data $x_{nd} \in \{0, 1, 2, \ldots, \infty\}$, we assume a Poisson likelihood model, i.e,

$$p(x_{nd}|\boldsymbol{\gamma}_{nd}) = \text{Poiss}(x_{nd}|\lambda_d(\mathbf{z}_n)) = \frac{(\lambda_d(\mathbf{z}_n))^{x_{nd}}\exp(-\lambda_d(\mathbf{z}_n))}{x_{nd}!}, \tag{11}$$

with $\boldsymbol{\gamma}_{nd} = \lambda_d(\mathbf{z}_n)$, where the mean parameter of the Poisson distribution corresponds to the output of a DNN.

4. Categorical data. For categorical data, codified using one-hot encoding, we assume a multinomial logit model such that the R-dimensional output of a DNN $\boldsymbol{\gamma}_{nd} = \{h_{d0}(\mathbf{z}_n), h_{d1}(\mathbf{z}_n), \ldots, h_{d(R-1)}(\mathbf{z}_n)\}$ represents the vector of unnormalized probabilities, such that the probability of every category is given by

$$p(x_{nd} = r|\boldsymbol{\gamma}_{nd}) = \frac{\exp(-h_{dr}(\mathbf{z}_n))}{\sum_{q=1}^{R}\exp(-h_{dq}(\mathbf{z}_n))}. \tag{12}$$

To ensure identifiability, we fix the value of $h_{d0}(\mathbf{z}_n)$ to zero.

5. Ordinal data. For ordinal data, codified using thermometer encoding,[1] we assume the ordinal logit model [39], where the probability of each (ordinal) category can be computed as

$$p(x_{nd} = r|\boldsymbol{\gamma}_{nd}) = p(x_{nd} \le r|\boldsymbol{\gamma}_{nd}) - p(x_{nd} \le r-1|\boldsymbol{\gamma}_{nd}), \tag{13}$$

with

$$p(x_{nd} \le r|\mathbf{z}_n) = \frac{1}{1+\exp(-(\theta_r(\mathbf{z}_n) - h_d(\mathbf{z}_n)))}. \tag{14}$$

Here, the thresholds $\theta_r(\mathbf{z}_n)$ divide the real line into R regions and $\mathbf{h}_d(\mathbf{z}_n)$ indicates the region (category) in which x_{nd} falls. Therefore, the likelihood parameters are $\boldsymbol{\gamma}_{nd} = \{h_d(\mathbf{z}_n), \theta_1(\mathbf{z}_n)\ldots, \theta_{R-1}(\mathbf{z}_n)\}$, which we model as the output of a DNN. To guarantee that $\theta_1(\mathbf{z}_n) < \theta_2(\mathbf{z}_n) < \ldots < \theta_{R-1}(\mathbf{z}_n)$, we apply a cumulative sum function to the $R-1$ positive real-valued outputs of the network.

Moreover, for all the likelihood parameters which need to be positive, we use the softplus function $f(x) = \log(1+\exp(x))$.

Remark. The caveat of the generative model in Fig. 1 is that we are losing the ability of deep neural networks to capture correlations among data attributes by amortizing the parameters, since we are learning a different network to link the latent variable $\mathbf{z}$ to each particular attribute by modeling the parameters of a certain observation model $p(\mathbf{x}_d|\mathbf{z})$. An alternative would be to use the approach in [40], where categorical one-hot encoded variables are approximated by continuous variables using jitter noise (uniform on [0,1]). When all attributes are assumed to be continuous, we could use a single network to map $\mathbf{z}$ to the parameters (mean and covariance) of a D-dimensional Gaussian distribution. However, this approach does not allow a combination of different likelihood models or distinguish categorical and ordinal data. In Section 3, we show how to solve this limitation by using a hierarchical model.

Handling heterogenous data ranges. Apart from different types of attributes, heterogeneous datasets commonly contain numerical attributes whose values correspond to completely different domains. For example, a dataset may contain the height of different individuals with values in the order of $1.5-2.0$ meters, and also their income, which might reach tens or even hundreds thousands of dollars per year. In order to learn the parameters of both the generative and the reconstruction models in Fig. 1, one might rely on stochastic gradient descent using at every iteration a minibatch estimate of the ELBO in (4).[2] However, the heterogenous nature of the data and these differences of value ranges between continuous variables result in broadly different likelihood parameters (e.g., the mean of the height is much lower than the mean of the income), leading in practice to heterogenous (and potentially unstable) gradient evaluations. To avoid the gradient evaluations of the ELBO being dominated by a subset of attributes, we apply a batch normalization layer at the input of the reconstruction model for the numerical variables, and we apply the complementary batch denormalization at the output layer of the generative model to denormalize the likelihood parameters.

In particular, for real-valued variables, we shift and scale the input data to the recognition model to ensure that the normalized minibatch has zero mean and variance equal to one. These shift and scale normalization parameters, μ' and σ', are afterwards used to denormalize the likelihood parameters of the Gaussian distribution, i.e., $x_{nd} \sim \mathcal{N}\big(x_{nd}|\sigma'\boldsymbol{\mu}_d(\mathbf{z}_n) + \mu', \sigma'^2\boldsymbol{\sigma}_d^2(\mathbf{z}_n)\big)$. For positive real-valued variables, for which a log-Normal model is used, we apply the same batch normalization at the encoder and denormalization at the decoder used for real-valued variables, but to the natural logarithm of the data, instead of directly to the data. We note that count variables are not batch denormalized at the decoder, but a normalized $\log(\cdot)$ transformation is used to feed the recognition network. With this batch normalization and denormalization layers at respectively the recognition and the generative models, we do not only enforce more stable evaluations (free of numerical errors) of the gradients, but we also speed-up the convergence of the optimization.

3. The heterogeneous-incomplete VAE (HI-VAE)

In the previous section, we have introduced a simple VAE architecture that handles incomplete and heterogeneous data. However, this approach might be too restrictive to capture complex and high-dimensional data. More specifically, we have assumed a standard Gaussian prior on the latent variables $\mathbf{z}_n$, which might be too restrictive based on the literature [6] and may be particularly problematic when the final goal is to estimate missing values in unstructured datasets (refer to the discussion under (8)). Similarly, we have assumed a generative model that fully factorizes for every (heterogenous) dimension in the data, losing the properties

[1] As an example, in an ordinal variable with three categories the lowest value is encoded as "100", the middle value as "110" and the highest value as "111".

[2] Although here we use the standard ELBO for VAEs, tighter log-likelihood lower bound, such as the one proposed in the importance weight encoder (IWAE) in [41], could also be applied.

Table 1
HI-VAE probabilistic model.

Generative	$p(\mathbf{x}_n, \mathbf{z}_n, \mathbf{s}_n) = p(\mathbf{s}_n)p(\mathbf{z}_n\|\mathbf{s}_n)\prod_d p(x_{nd}\|\boldsymbol{\gamma}_{nd} = h_d(\mathbf{y}_{nd}, \mathbf{s}_n))$, where where $\mathbf{Y}_n = [\mathbf{y}_{n1}, \dots, \mathbf{y}_{nD}] = \mathbf{g}(\mathbf{z}_n)$ $p(\mathbf{s}_n) = \text{Categorical}(\mathbf{s}_n\|\boldsymbol{\pi})$, $p(\mathbf{z}_n\|\mathbf{s}_n) = \mathcal{N}(\mathbf{z}_n\|\boldsymbol{\mu}_p(\mathbf{s}_n), \mathbf{I}_K)$
Recognition	$q(\mathbf{s}_n, \mathbf{z}_n, \mathbf{x}_n^m\|\mathbf{x}_n^o) = q(\mathbf{s}_n\|\mathbf{x}_n^o)q(\mathbf{z}_n\|\mathbf{x}_n^o, \mathbf{s}_n)\prod_{d\in\mathcal{M}_n} p(x_{nd}\|\mathbf{z}_n, \mathbf{s}_n)$, where $q(\mathbf{s}_n\|\mathbf{x}_n^o) = \text{Categorical}(\mathbf{s}_n\|\boldsymbol{\pi}(\tilde{\mathbf{x}}_n))$ $q(\mathbf{z}_n\|\mathbf{x}_n^o, \mathbf{s}_n) = \mathcal{N}(\mathbf{z}_n\|\boldsymbol{\mu}_q(\tilde{\mathbf{x}}_n, \mathbf{s}_n), \boldsymbol{\Sigma}_q(\tilde{\mathbf{x}}_n, \mathbf{s}_n))$
ELBO	$\log p(\mathbf{X}^o) \geq \sum_{n=1}^N \left(\mathbb{E}_{q(\mathbf{s}_n,\mathbf{z}_n\|\mathbf{x}_n^o)}\left[\sum_{d\in\mathcal{O}_n}\log p(x_{nd}\|\mathbf{z}_n, \mathbf{s}_n)\right]\right)$ $-\sum_{n=1}^N \mathbb{E}_{q(\mathbf{s}_n\|\mathbf{x}_n^o)}[KL(q(\mathbf{z}_n\|\mathbf{x}_n^o, \mathbf{s}_n)\|\|p(\mathbf{z}_n\|\mathbf{s}_n))]$ $-\sum_{n=1}^N KL(q(\mathbf{s}_n\|\mathbf{x}_n^o)\|\|p(\mathbf{s}_n))$
Likelihoods	Real-valued data (Normal): $\boldsymbol{\gamma}_{nd} = \{\mu_d(\mathbf{y}_{nd}, \mathbf{s}_n), \sigma_d^2(\mathbf{s}_n)\}$ Positive real-valued data (log-Normal): $\boldsymbol{\gamma}_{nd} = \{\mu_d(\mathbf{y}_{nd}, \mathbf{s}_n), \sigma_d^2(\mathbf{s}_n)\}$ Count data (Poisson): $\boldsymbol{\gamma}_{nd} = \lambda_d(\mathbf{y}_{nd}, \mathbf{s}_n)$ Categorical (Mult. logit): $\boldsymbol{\gamma}_{nd} = \{h_{d0}(\mathbf{y}_{nd}, \mathbf{s}_n), \dots, h_{d(R-1)}(\mathbf{y}_{nd}, \mathbf{s}_n)\}$ Ordinal (Ordinal logit): $\boldsymbol{\gamma}_{nd} = \{h_d(\mathbf{y}_{nd}, \mathbf{s}_n), \theta_1(\mathbf{s}_n)\dots, \theta_{R-1}(\mathbf{s}_n)\}$

of an amortized generative model where the different dimensions share the weights of a common DNN capturing the relationships between attributes (as CNNs capture correlations between pixels in an image). In this section, we overcome these limitations of the model discussed in the previous section and remark that the models proposed in this paper are, in fact, compatible with the current developments in VAE literature.

In order to prevent the KL term in (4) from dominating the ELBO, thus penalizing rich posterior distributions for $\mathbf{z}_n$, we can impose structure in the latent variable representation $\mathbf{z}_n$ through its prior distribution. We propose a Gaussian mixture prior $p(\mathbf{z}_n)$ [42], such that

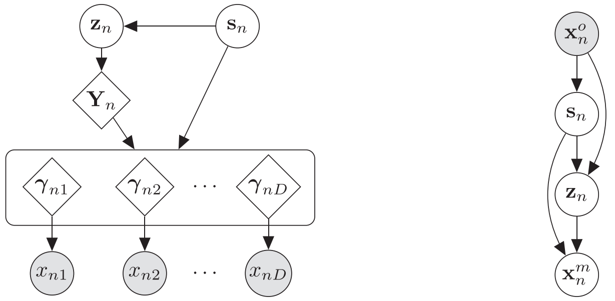
$$p(\mathbf{s}_n) = \text{Categorical}(\mathbf{s}_n|\boldsymbol{\pi}) \tag{15}$$

$$p(\mathbf{z}_n|\mathbf{s}_n) = \mathcal{N}(\mathbf{z}_n|\boldsymbol{\mu}_p(\mathbf{s}_n), \mathbf{I}_K), \tag{16}$$

where $\mathbf{s}_n$ is a one-hot encoding vector representing the component in the mixture, i.e., the mean of the Gaussian component that generates $\mathbf{z}_n$. For simplicity, we assume a uniform Gaussian mixture with $\pi_\ell = 1/L$ for all ℓ.

Moreover, to allow the model to more accurately capture the statistical dependencies among heterogeneous attributes, we propose a hierarchical structure that allows different attributes to share network parameters (i.e., to amortize the generative model). More specifically, we introduce an intermediate homogenous representation of the data $\mathbf{Y} = [\mathbf{y}_{n1}, \dots, \mathbf{y}_{nD}]$, which is jointly generated by a single DNN with input $\mathbf{z}_n$, $\mathbf{g}(\mathbf{z}_n)$. Then, the likelihood parameters of each attribute d are the output of an independent DNN with inputs $\mathbf{y}_{nd}$ and $\mathbf{s}_n$, such that $p(x_{nd}|\boldsymbol{\gamma}_{nd} = \mathbf{h}_d(\mathbf{y}_{nd}, \mathbf{s}_n))$. Note that, in this hierarchical structure, the top level (from $\mathbf{z}_n$ to $\mathbf{Y}_n$) captures statistical dependencies among the attributes through the shared DNN $\mathbf{g}(\mathbf{z}_n)$, while the bottom level in the hierarchy (from $\mathbf{Y}_n$ and $\mathbf{s}_n$ to $\mathbf{x}_n$) accounts for heterogeneous likelihood models using d independent DNNs $\mathbf{h}_d(\mathbf{y}_{nd}, \mathbf{s}_n)$. The resulting generative model, that is hereafter referred to as Heterogeneous-Incomplete VAE (HI-VAE), is shown in Fig. 2 and is formulated as indicated in Table 1, which also shows how we parametrize in the HI-VAE the different likelihood models provided in Section 2.2.[3]

Regarding the recognition network (Fig. 2b) $q(\mathbf{s}_n|\mathbf{x}_n^o)$ is a categorical distribution with a parameter vector $\boldsymbol{\pi}$ given by the output of a DNN with input $\tilde{\mathbf{x}}_n$ and a soft-max output function. Then, a concatenation of both $\mathbf{s}_n$ and $\tilde{\mathbf{x}}_n$ is used to construct the moments of the Gaussian $q(\mathbf{z}_n|\mathbf{x}_n^o, \mathbf{s}_n)$ posterior distribution via two independent NNs. Finally, to enforce that the model captures all correlations using the hidden variables $\mathbf{s}_n$ and $\mathbf{z}_n$, in the recognition network we assume that the posterior distribution of the missing attributes $\mathbf{x}_n^m$ is conditionally independent on the observed attributes $\mathbf{x}_n^o$, given $\mathbf{s}_n$ and $\mathbf{z}_n$. Hence, our variational distribution (or, equivalently, our recognition model) factorizes as:



(a) Generative Model (b) Recognition Model

Fig. 2. Graphical models for the generative and recognition models of the HI-VAE.

$$q(\mathbf{s}_n, \mathbf{z}_n, \mathbf{x}_n^m|\mathbf{x}_n^o) = q(\mathbf{s}_n|\mathbf{x}_n^o)q(\mathbf{z}_n|\mathbf{x}_n^o, \mathbf{s}_n)\prod_{d\in\mathcal{M}_n} p(x_{nd}|\mathbf{z}_n, \mathbf{s}_n).$$

By combining the HI-VAE generative model and the proposed recognition network, we derive the expression for ELBO in Table 1, where the Gumbel-softmax reparameterization trick [42] is used to draw differentiable samples from $q(\mathbf{s}_n, \mathbf{z}_n|\mathbf{x}_n^o)$.

4. Experiments

In this section, we first evaluate the performance of the HI-VAE at solving a missing data imputation task in heterogeneous data, comparing it to other methods in the literature. Then, we focus on a classification task, where we evaluate the classification degradation due to performing mean imputation for the missing data in supervised models compared to using the fully generative HI-VAE, which does not require data imputation. The models and datasets employed in the experiments can be found in the following public repository https://github.com/probabilistic-learning/HI-VAE.

4.1. Missing data imputation

In our first experiment, we evaluate the performance of the proposed HI-VAE at imputing missing data. We use six different heterogenous datasets from the UCI repository [43], which vary both in the number of instances and attributes, as well as in the statistical data types of the attributes. We summarize the main characteristics of these databases in Table 2. For each dataset we generate 10 different incomplete datasets, removing completely at random a percentage of the data ranging from a 10% deletion to a 50%.

Imputation strategy. Once the HI-VAE model is trained, the imputation of missing data is performed in a three-step process: First,

[3] Other likelihood functions (e.g., a Gamma distribution) and data types (e.g., interval data using e.g., a Beta distribution) can be readily be incorporated.

Table 2
Dataset description. The attributes include the target variable for those datasets that have an associated binary classification task.

Database	Objects	Attributes	# Real	# Positive	# Categorical	# Ordinal	# Count
Adult	32,561	12	0	3	6	1	2
Breast	699	10	0	0	1	9	0
Default Credit	30,000	24	6	7	4	6	1
Letter	20,000	17	0	0	1	16	0
Spam	4601	58	0	57	1	0	0
Wine	6497	13	0	11	1	0	1

we perform the MAP estimate of $q(\mathbf{z}_n, \mathbf{s}_n|\mathbf{x}_n^o)$ to obtain $\hat{\mathbf{z}}_n$ and $\hat{\mathbf{s}}_n$. With these MAP estimates, we evaluate the generative model, obtaining $\hat{\mathbf{Y}}_n = \mathbf{g}(\hat{\mathbf{z}}_n)$ and $\hat{\gamma}_{nd} = \mathbf{h}_d(\hat{\mathbf{y}}_{nd}, \hat{\mathbf{s}}_n)$ for every attribute. Finally, the imputed values $\hat{\mathbf{x}}_n$ are obtained as the mode of each distribution $p(x_{nd}|\hat{\gamma}_{nd})$, where the computation of the mode depends on the likelihood model of the attribute. A further discussion on imputation methods is provided in Section 4.1.1.

Imputation error. We compare the above models in terms of average imputation error computed as $\text{AvgErr} = 1/D\sum_d \text{err}(d)$, where we use the following error metrics for each attribute, since the computation of the errors depends on the type of variable we are considering:

- Normalized Root Mean Square Error (NRMSE) for numerical variables, i.e.,

$$\text{err}(d) = \frac{\sqrt{1/n\sum_n (x_{nd} - \hat{x}_{nd})^2}}{\max(\mathbf{x}_d) - \min(\mathbf{x}_d)}. \tag{17}$$

- Accuracy error for categorical variables, i.e.,

$$\text{err}(d) = \frac{1}{n}\sum_n I(x_{nd} \neq \hat{x}_{nd}). \tag{18}$$

- Displacement error for ordinal variables, i.e.,

$$\text{err}(d) = \frac{1}{n}\sum_n |\frac{x_{nd} - \hat{x}_{nd}}{R}|. \tag{19}$$

Comparison. We compare the performance of the following methods for missing data imputation:

- **Mean Imputation**: We use as baseline an algorithm that imputes the mean of each continuous attribute and the mode of each discrete attribute.
- **MICE:** Multiple Imputation by Chained Equations [44], which is an iterative method that performs a series of supervised regression models, in which missing data is modeled conditional upon the other variables in the data, including those imputed in previous rounds of the algorithm. We use MICE implementation within the *fancyimpute* package https://github.com/iskandr/fancyimpute, which, in its current implementation, only allows the user to pick a homogeneous regression model for all attributes, independently of whether they are numerical or nominal.
- **GLFM:** General latent feature model for heterogeneous data [1], which was initially introduced for table completion in heterogeneous datasets in [45]. This method handles all the numerical and nominal data types described in Section 1 and performs MCMC inference. We run 5000 iterations of the sampler using the available implementation in https://github.com/ivaleraM/GLFM.
- **GAIN:** Generative adversarial network for missing data imputation [34], which uses MSE as a loss function for numerical variables, and cross-entropy for binary variables. We train GAIN for 2000 epochs using the network specifications and hyperparameters reported in [34].
- **HI-VAE:** Model introduced in Section 3, which we implement in TensorFlow using only one dense layer for all the parameters of the encoder and decoder of the HI-VAE). We set the dimensionality of **z, y** and **s** to 10, 5 and 10, respectively. The parameter τ of the Gumbel-Softmax is annealed using a linear decreasing function on the number of epochs, from 1 to 10^{-3}. We train our algorithms for 2000 epochs using minibatches of 1000 samples. We note that we have used the same NN architecture in all experiments and, therefore, further improvements could be achieved by cross-validating the architecture for each database. We further explore this aspect in Section 4.1.1.

4.1.1. Variations in the HI-VAE design

First, we explore different aspects of the design (such as network architecture, normalization layer, and hyperparameter selection) and how the use of the missing data imputation strategy of HI-VAE may improve the performance of the proposed HI-VAE for missing data estimation.

Network design. Here we analyze the sensitivity of the HI-VAE to the network architecture. To this end, we vary dimensionality of **s, z** and **y** and consider both generator and inference networks with either one or two dense layers with ReLu activation functions. In Fig. 3 we show the HI-VAE average imputation error with a 20% missing data rate for different network configurations and latent space dimensions. Here we observe that, while using two layers and a larger latent dimension (brown bars) tend to improve the performance, significant gains are only observed with Letter database, where more complex architectures lead to a lower imputation error.

HI-VAE imputation strategy. Once we have trained the generative model, to impute missing data we can either sample from the generative model or use the inferred parameters of the output distribution, e.g., impute the mode of the inferred distribution. To illustrate the differences, we show in Figs. 4 and 5 the goodness of fit provided by the HI-VAE and the GLFM in a positive real-valued variable and a categorical variable with 6 categories, both belonging to the Adult dataset. Specifically, we show (top row) the true distribution of the data together with the HI-VAE output distribution for the observed data and the HI-VAE output distribution for the missing values. We show results for HI-VAE using the mode of the distribution and HI-VAE using one sample for imputation. We also show results for the GLFM. Further, in the bottom row we show the Q-Q plot for the positive-real variable and the confusion matrix for the categorical variable (see the figure caption for more details). Note that, while both the HI-VAE and the GLFM result in a good fit of the positive real variable (although the HI-VAE provides a smoother, and thus, more realistic distribution for the data); the GLFM fails at capturing the categorical variable–it assigns all the probability to a single category. These results are consistent with Table 4 in the paper, which demonstrate the superior ability of the HI-VAE to perform missing data imputation in nominal variables.

Normalization layer. Finally, we study the effect of the batch (de-)normalization layer at the input of the reconstruction (and at the output layer of the generative) model for the numerical vari-

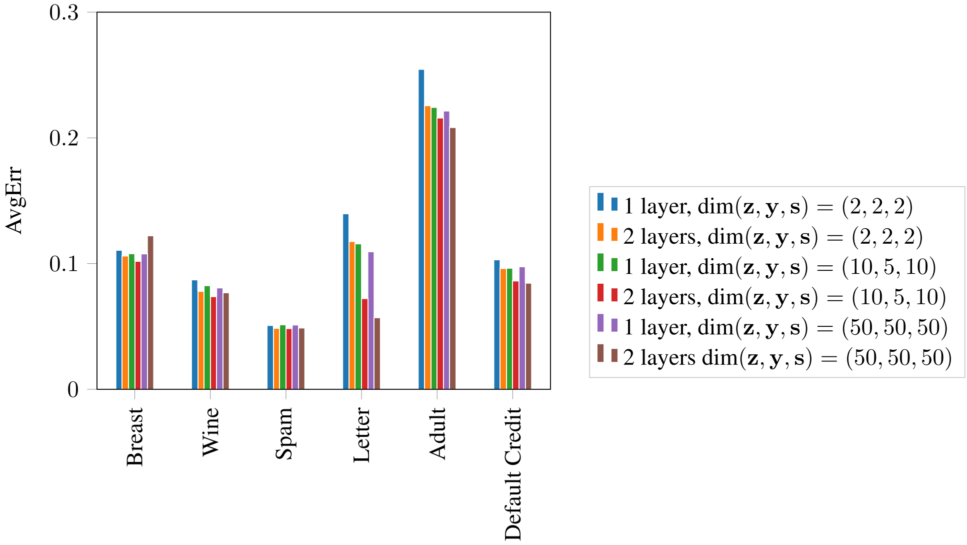


Fig. 3. HI-VAE average imputation error for different network configurations and latent space dimensions with a 20% rate of missing data.

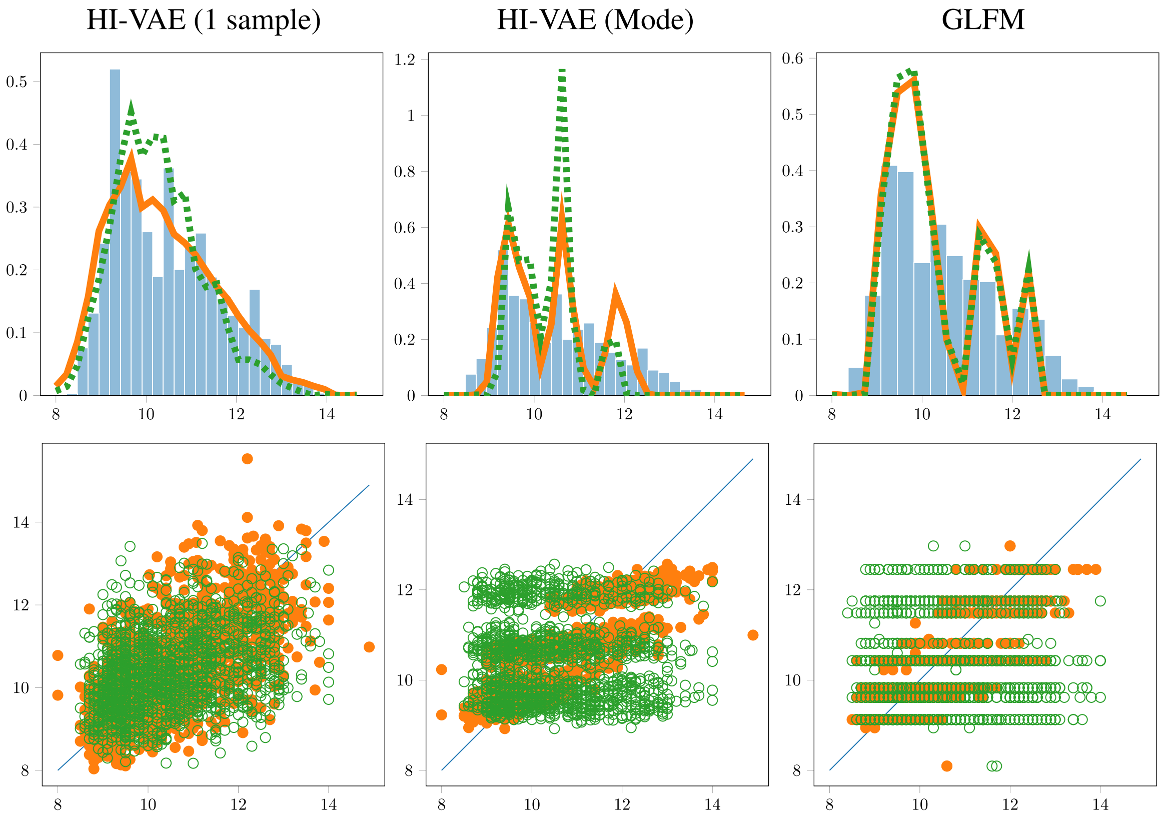


Fig. 4. We demonstrate the fit provided by the HI-VAE and the GLFM in a positive real-valued variable of the Adult dataset. Top row depicts the true empirical data distribution (shadowed histogram) and the inferred data distribution for the observed attributes in dashed line and for the missing data in solid line. The bottom row shows the Q-Q plot (observed in orange (•) marker and missing in green (◦) marker). The left-most column shows the results for the HI-VAE when we sample from the model posterior distribution (given the observed data) to impute, while for the center column we use the mode of the posterior. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

ables described in Section 2.2. The first two rows of Tables 3 and 4 show the imputation error of the HI-VAE with and without the (de-)normalization layers for all the considered datasets containing numerical variables (since the normalization layer only apply to numerical variables), and for a 20% of missing data selected completely at random. Here we observe that the normalization layer not only leads to a significant improvement in terms of imputation error for the Adult, the Spam and the Wine datasets, but it also prevents numerical errors from occurring during inference – for the Default dataset, the gradients of the ELBO with respect to the model parameters take infinite values.

4.1.2. Comparison with exiting methods

Finally, we compare the performance of the HI-VAE with existing methods in the literature to input missing data in heterogenous datasets. For the HI-VAE we use here a relatively-simple configura-

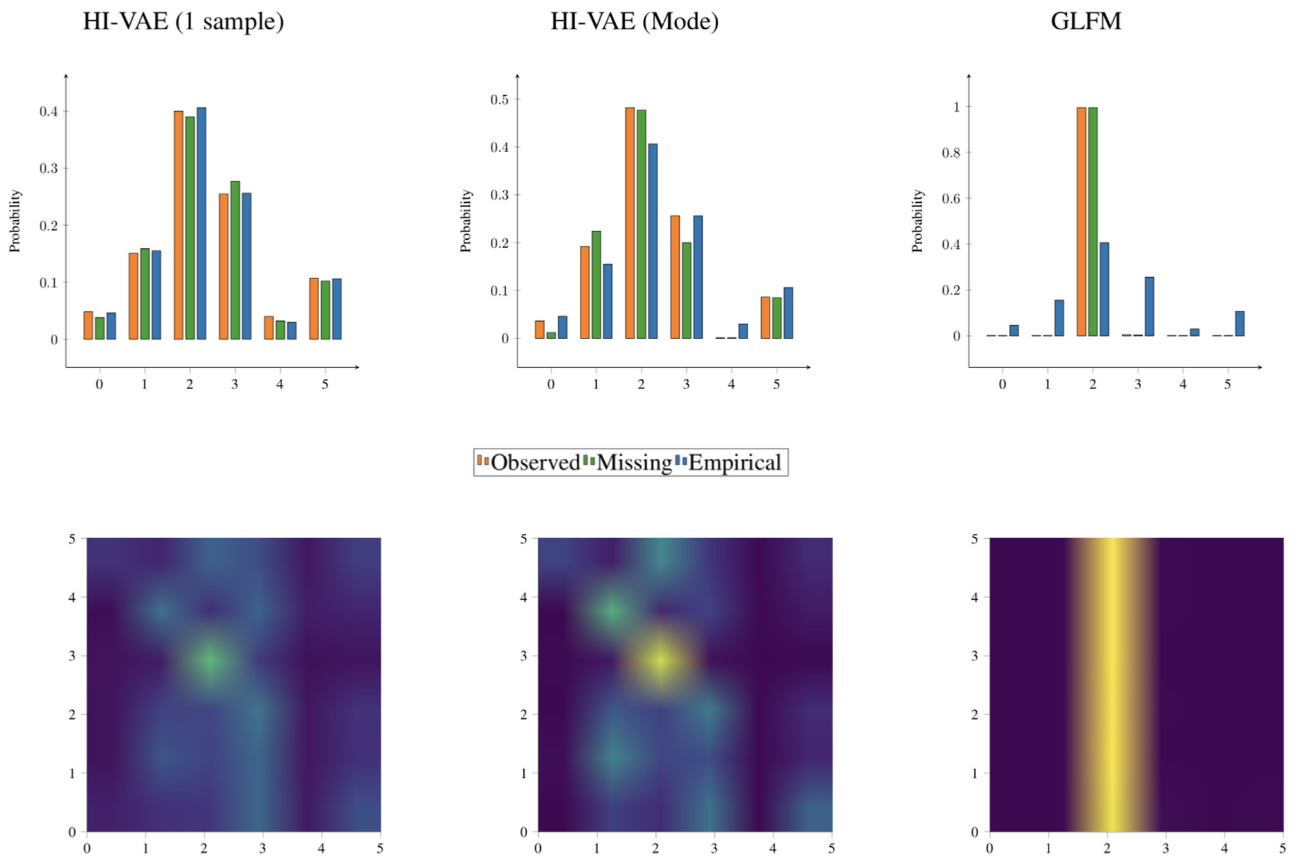


Fig. 5. We demonstrate the fit provided by the HI-VAE and the GLFM in a categorical variable with 6 categories of the Adult dataset. Top row depicts the true empirical data distribution and the inferred data distribution for the observed attributes and for the missing data. The bottom row shows the missing data confusion matrix.

Table 3
Imputation error. Average and standard deviation of the imputation error for a 20% of missing data, evaluated exclusively over **numeric variables**.

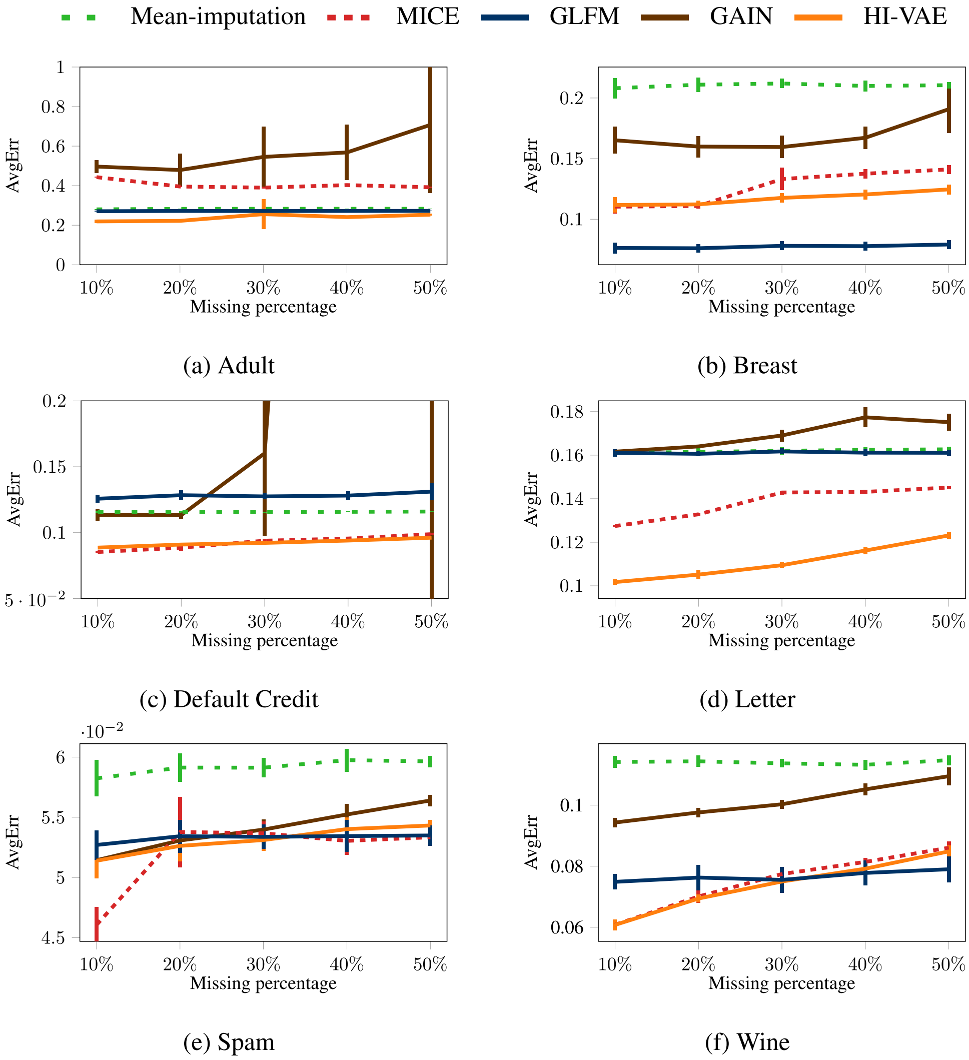
Model	Adult	Breast	DefaultCredit	Letter	Spam	Wine
HI-VAE (no norm.)	0.210 ± 0.028	–	*Inf*	–	0.054 ± 0.018	0.165 ± 0.042
HI-VAE	0.106 ± 0.002	–	0.043 ± 0.001	–	0.052 ± 0.001	**0.074 ± 0.001**
Mean imputation	0.111 ± 0.002	–	0.056 ± 0.001	–	0.053 ± 0.001	0.103 ± 0.002
MICE	0.108 ± 0.002	–	**0.035 ± 0.002**	–	0.052 ± 0.003	0.074 ± 0.002
GLFM	**0.083 ± 0.001**	–	0.051 ± 0.005	–	0.052 ± 0.001	0.082 ± 0.004
GAIN	0.225 ± 0.192	–	0.044 ± 0.002	–	**0.049 ± 0.001**	0.086 ± 0.002

Table 4
Imputation error. Average and standard deviation of the imputation error for a 20% of missing data, evaluated exclusively over **nominal variables**.

Model	Adult	Breast	DefaultCredit	Letter	Spam	Wine
HI-VAE (no norm.)	0.406 ± 0.005	–	0.202 ± 0.003	–	0.166 ± 0.019	0.245 ± 0.017
HI-VAE	**0.304 ± 0.006**	0.112 ± 0.003	**0.158 ± 0.001**	**0.105 ± 0.002**	**0.111 ± 0.009**	0.016 ± 0.003
Mean imputation	0.405 ± 0.002	0.211 ± 0.006	0.2 ± 0.001	0.162 ± 0.002	0.393 ± 0.014	0.248 ± 0.014
MICE	0.601 ± 0.002	0.111 ± 0.002	0.163 ± 0.003	0.133 ± 0.0	0.168 ± 0.012	0.02 ± 0.004
GLFM	0.407 ± 0.003	**0.076 ± 0.003**	0.236 ± 0.012	0.161 ± 0.001	0.154 ± 0.02	**0.006 ± 0.001**
GAIN	0.66 ± 0.025	0.16 ± 0.009	0.211 ± 0.005	0.164 ± 0.001	0.276 ± 0.017	0.236 ± 0.014

tion with a single dense layer with a latent space of dimensions $\dim(\mathbf{z}) = 10, \dim(\mathbf{s}) = 10$ and $\dim(\mathbf{y}) = 10$. A more careful design of the HI-VAE structural parameters for each dataset may thus improve the HI-VAE performance. Fig. 6 summarizes the average imputation error for each database as we vary the fraction of missing data. The results clearly show that the proposed HI-VAE is the only method that consistently outperforms mean imputation in all the datasets—since mean imputation assumes all the attributes to be independent, any missing data imputation method that accounts for statistical dependencies in the data should perform at least as accurately as mean imputation. The second more robust model is the GLFM, which performs best in small datasets (Breast and Wine). This might be explained by the fact that, while it accounts for mixed nominal and discrete data, it relies on Gibbs-sampling for inference, scaling and mixing poorly for larger datasets. In contrast, the MICE and GAIN[4] are outperformed by the Mean-

[4] We would like to clarify that the reported results do not quite match those provided in [34], despite using the code and the hyperparameters provided by the



(a) Adult

(b) Breast

(c) Default Credit

(d) Letter

(e) Spam

(f) Wine

Fig. 6. *Missing Data.* Average imputation error for different percentages of missing data (completely at random).

imputation baseline in several datasets, most likely, due to the fact that they do not account for different types of mixed nominal and numerical attributes.

A deeper understanding of the results in Fig. 6 can be obtained by separately analyzing the error in numeric variables (real, positive and count variables) in Table 3, and nominal variables (categorical/ordinal variables) in Table 4. In both cases, we use 20% missing data. While for numeric variables HI-VAE achieves a comparable error w.r.t. the rest of the methods, it is in the imputation of nominal variables where HI-VAE achieves a remarkable gain, being the best performing method in four out of six cases. These results demonstrate the superior ability of HI-VAE to exploit underlying correlations among the set of heterogeneous attributes.

4.2. Predictive task

Although the HI-VAE is a fully unsupervised generative model, we evaluate its performance at solving a classification task, a multi-class classification problem for the Letter dataset (with 26 classes corresponding to the different letters) and a binary classification problem for the rest of databases (predicting the binary label of each element of the dataset). The idea behind this experiment, is to treat the classes to be predicted as missing entries in the target attribute, using HI-VAE to provide an imputation of these missing entries. We use 50% of the data for training, which for HI-VAE means that we remove 50% of the labels in the target attribute to train the generative model. Regarding the training data, we consider three different scenarios: the first assumes complete input attributes in the training set (no missing data), the second assumes 10% missing values in the input training data, and the third assumes 50% missing values. Since the supervised methods we compare HI-VAE to cannot handle missing data, we impute the mean (or the mode for discrete attributes) of each attribute to the missing input values during training. Here, we compare our HI-VAE with two supervised methods: deep logistic regression (DLR) and the conditional VAE (CVAE) in [32]. Following our results in Fig. 3, we use the basic configuration for the HI-VAE, i.e., one dense layer and **z, y** and **s** to 10, 5 and 10, respectively, for all datasets except for the Letter, for which we use two dense layers with ReLU activations and 50-dimensional latent spaces.

Results. Table 5 summarizes the results, where we observe that our HI-VAE method provides competitive results in all cases. Furthermore, note HI-VAE provides the best results for both Wine and Breast, while showing less degradation with increasing fraction of missing input data in the DefaultCredit and Spam. These results

authors. For the sake of reproducibility, we will incorporate the GAIN implementation to our public repository.

Table 5
Prediction Accuracy. Average and standard deviation of the classification error when we use 50% of the labels for training and assume complete input data and 10% and 50% of missing values in input data (right-hand table).

% Missing	Model	Breast	DefaultCredit	Letter	Spam	Wine
0%	DLR	0.041 ± 0.01	**0.179 ± 0.002**	0.142 ± 0.003	**0.081 ± 0.005**	0.018 ± 0.003
	CVAE	0.04 ± 0.012	**0.179 ± 0.001**	0.14 ± 0.004	0.081 ± 0.006	0.016 ± 0.002
	HIVAE	**0.026 ± 0.005**	0.2 ± 0.004	**0.117 ± 0.014**	0.096 ± 0.007	**0.014 ± 0.002**
10%	DLR	0.04 ± 0.009	**0.184 ± 0.001**	0.229 ± 0.002	0.09 ± 0.005	0.027 ± 0.003
	CVAE	0.048 ± 0.009	0.184 ± 0.002	0.227 ± 0.003	**0.088 ± 0.006**	0.025 ± 0.003
	HIVAE	**0.031 ± 0.007**	0.201 ± 0.002	**0.212 ± 0.017**	0.103 ± 0.008	**0.022 ± 0.006**
50%	DLR	0.08 ± 0.014	**0.196 ± 0.003**	**0.496 ± 0.005**	**0.134 ± 0.008**	0.078 ± 0.006
	CVAE	0.101 ± 0.038	0.197 ± 0.003	**0.496 ± 0.005**	0.138 ± 0.009	0.078 ± 0.005
	HIVAE	**0.052 ± 0.012**	0.205 ± 0.003	0.589 ± 0.014	0.138 ± 0.005	**0.042 ± 0.005**

show that a fully generative model might be preferred over a supervised model with imputed data.

5. Conclusions

In this paper, we focus on designing and inferring deep generative models (in particular, VAEs) for heterogeneous and incomplete data. We note that it is not a straightforward problem, and that it has been overlooked in the literature. The main issues covered in this paper and for which HI-VAE provides an effective solution can be summarized as follow: First, standard (and conditional) VAEs assume complete data during training, however, missing data imputation is a fully unsupervised task where missing values may appear ubiquitously in the dataset. Unfortunately, while VAEs perform accurate inference through a recognition model sharing parameters among inputs, this is not directly possible when training data is incomplete (DNNs require complete input). In HI-VAE, we derive a lower-bound on the data marginal likelihood that depends exclusively on the observed data. Also, we propose methods to deal with missing values in the recognition network. Second, when data are heterogeneous in both statistical types and ranges, the inference of a joint set of parameters that accurately captures the statistical dependencies among attributes results in a complex optimization problem with many local optima. Intuitively, each local optima potentially captures the correlations between a subset of attributes and treats the rest as independent, while the global optima captures all the existing correlations in the data. In HI-VAE, we enforce correlation by using a joint DNN to construct the parameters that define the output distribution of each of the attributes. Third, in contrast to deep generative approaches for structured and homogeneous data (e.g., images or text), the use of more complex DNNs (e.g., CNNs or RNNs) does not necessarily lead to a better fitting of the data in heterogeneous datasets, where there is no clear notion of correlation to be exploited by weight sharing of the DNNs. The hierarchical HI-VAE generative model captures correlation among the different attributes by using a latent space spanned by a Gaussian mixture.

Our empirical results show that our proposed HI-VAE outperforms competitors on a heterogenous data completion task and provides comparable results in classification accuracy to deep supervised methods, which cannot handle missing values in the input data, therefore, requiring imputation of missing inputs in the data. Future work includes the extension to more complex attributes such as images or text, and the generalization to temporal heterogeneous series with missing data.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors wish to thank Christopher K. I. Williams, for fruitful discussions and helpful comments to the manuscript. Alfredo Nazabal would like to acknowledge the funding provided by the UK Government's Defence & Security Programme in support of the Alan Turing Institute, EPSRC Grant EP/N510129/1. The work of Pablo M. Olmos is sup-ported by Spanish government MCI under grant RTI2018-099655-B-100, by Comunidad de Madrid under grants IND2017/TIC-7618, IND2018/TIC-9649, and Y2018/TCS-4705, by BBVA Foundation under the Deep-DARWiNproject, and by the European Union (FEDER and the European Research Council (ERC) through the European Unions Horizon 2020 research and innovation program under Grant 714161). Zoubin Ghahramani acknowledges support from the Alan Turing Institute (EPSRC Grant EP/N510129/1) and EPSRC Grant EP/N014162/1, and donations from Google and Microsoft Research. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References

[1] I. Valera, Z. Ghahramani, Automatic discovery of the statistical types of variables in a dataset, in: Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, vol. 70, 2017, pp. 3521–3529.
[2] D.P. Kingma, M. Welling, Auto-encoding variational bayes, in: Proceedings of the 2nd International Conference on Learning Representations (ICLR), Banff, Canada, 2014.
[3] D. Rezende, S. Mohamed, Variational inference with normalizing flows, in: Proceedings of the 32nd International Conference on Machine Learning, Lille, France, vol. 37, 2015, pp. 1530–1538.
[4] C. Li, J. Zhu, B. Zhang, Max-margin deep generative models for (semi-)supervised learning, IEEE Trans. Pattern Anal. Mach. Intell. 40 (11) (2018) 2762–2775.
[5] L. Mescheder, S. Nowozin, A. Geiger, Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks, in: Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, vol. 70, 2017, pp. 2391–2400.
[6] J. Tomczak, M. Welling, VAE with a VampPrior, in: Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, Playa Blanca, Lanzarote, Canary Islands, vol. 84, 2018, pp. 1214–1223.
[7] S.K. Ainsworth, N.J. Foti, A.K.C. Lee, E.B. Fox, oi-VAE: output interpretable VAEs for nonlinear group factor analysis, in: Proceedings of the 35th International Conference on Machine Learning, Stockholm Sweden, vol. 80, 2018, pp. 119–128.
[8] A. Vahdat, W. Macready, Z. Bian, A. Khoshaman, E. Andriyash, DVAE++: Discrete variational autoencoders with overlapping transformations, in: Proceedings of the 35th International Conference on Machine Learning, Stockholm Sweden, vol. 80, 2018, pp. 5035–5044.
[9] C. Nash, C.K.I. Williams, The shape variational autoencoder: a deep generative model of part-segmented 3D objects, Comput. Graphics Forum 36 (5) (2017) 1–12.
[10] S. Nowozin, B. Cseke, R. Tomioka, f-GAN: training generative neural samplers using variational divergence minimization, in: Advances in Neural Information Processing Systems (NIPS) 29, 2016, pp. 271–279.
[11] L.Q. Tran, X. Yin, X. Liu, Representation learning by rotating your faces, IEEE Transactions on Pattern Analysis and Machine Intelligence (2019). (Early access)
[12] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A.C. Courville, Improved training of Wasserstein GANs, in: Advances in Neural Information Processing Systems (NIPS) 30, 2017, pp. 5767–5777.

[13] Y. Li, K. Swersky, R. Zemel, Generative moment matching networks, in: Proceedings of the 32nd International Conference on Machine Learning, Lille, France, vol. 37, 2015, pp. 1718–1727.
[14] X. Chen, C. Xu, X. Yang, L. Song, D. Tao, Gated-GAN: adversarial gated networks for multi-collection style transfer, IEEE Trans. Image Process. 28 (2) (2019) 546–560.
[15] I.O. Tolstikhin, S. Gelly, O. Bousquet, C.-J. Simon-Gabriel, B. Schölkopf, AdaGAN: Boosting generative models, in: Advances in Neural Information Processing Systems (NIPS) 30, 2017, pp. 5424–5433.
[16] Y. Mroueh, T. Sercu, V. Goel, McGan: Mean and covariance feature matching GAN, in: Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, vol. 70, 2017, pp. 2527–2535.
[17] A. Farhangfar, L. Kurgan, J. Dy, Impact of imputation of missing values on classification error for discrete data, Pattern Recognit. 41 (12) (2008) 3692–3705.
[18] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, X. Chen, Improved techniques for training GANs, in: Advances in Neural Information Processing Systems (NIPS) 29, 2016, pp. 2234–2242.
[19] L. Zhao, H. Bai, J. Liang, B. Zeng, A. Wang, Y. Zhao, Simultaneous color-depth super-resolution with conditional generative adversarial networks, Pattern Recognit. 88 (2019) 356–369.
[20] Z. Yang, Z. Hu, R. Salakhutdinov, T. Berg-Kirkpatrick, Improved variational autoencoders for text modeling using dilated convolutions, in: Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, vol. 70, 2017, pp. 3881–3890.
[21] Y. Cao, L. Jia, Y. Chen, N. Lin, C. Yang, B. Zhang, Z. Liu, X. Li, H. Dai, Recent advances of generative adversarial networks in computer vision, IEEE Access 7 (2019) 14985–15006.
[22] S. Yu, R. Liao, W. An, H. Chen, E.B. GarcÃa, Y. Huang, N. Poh, Gaitganv2: invariant gait feature extraction using generative adversarial networks, Pattern Recognit. 87 (2019) 179–189.
[23] A. Atapour-Abarghouei, S. Akcay, G.P. de La Garanderie, T.P. Breckon, Generative adversarial framework for depth filling via Wasserstein metric, cosine transform and domain transfer, Pattern Recognit. 91 (2019) 232–244.
[24] M. Freitag, S. Amiriparian, S. Pugachevskiy, N. Cummins, B. Schuller, auDeep: unsupervised learning of representations from audio with deep recurrent neural networks, J. Mach. Learn. Res. 18 (1) (2017) 6340–6344.
[25] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, J. Mach. Learn. Res. 17 (59) (2016) 1–35.
[26] T. Kim, M. Cha, H. Kim, J.K. Lee, J. Kim, Learning to discover cross-domain relations with generative adversarial networks, in: Proceedings of the 34th International Conference on Machine Learning, International Convention Centre, Sydney, Australia, vol. 70, 2017, pp. 1857–1865.
[27] Y. Taigman, A. Polyak, L. Wolf, Unsupervised cross-domain image generation, in: International Conference on Learning Representations (ICLR), Toulon, France, 2017.
[28] M.-Y. Liu, T. Breuel, J. Kautz, Unsupervised image-to-image translation networks, in: Advances in Neural Information Processing Systems (NIPS) 30, 2017, pp. 700–708.
[29] J. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2242–2251.
[30] L. Castrejón, Y. Aytar, C. Vondrick, H. Pirsiavash, A. Torralba, Learning aligned cross-modal representations from weakly aligned data, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2940–2949.
[31] M. Liu, J. Shi, K. Cao, J. Zhu, S. Liu, Analyzing the training processes of deep generative models, IEEE Trans. Vis. Comput. Graph. 24 (1) (2018) 77–87.
[32] K. Sohn, H. Lee, X. Yan, Learning structured output representation using deep conditional generative models, in: Advances in Neural Information Processing Systems (NIPS) 28, 2015, pp. 3483–3491.
[33] D.P. Kingma, S. Mohamed, D. Jimenez Rezende, M. Welling, Semi-supervised learning with deep generative models, in: Advances in Neural Information Processing Systems (NIPS) 27, 2014, pp. 3581–3589.
[34] J. Yoon, J. Jordon, M. van der Schaar, GAIN: Missing data imputation using generative adversarial nets, in: Proceedings of the 35th International Conference on Machine Learning, Stockholm Sweden, vol. 80, 2018, pp. 5689–5698.
[35] D.P. Kingma, M. Welling, Do GANs actually learn the distribution? An empirical study, in: Proceedings of the 6th International Conference on Learning Representations (ICLR), Vancouver, Canada, 2018.
[36] D.B. Rubin, Inference and missing data, Biometrika 63 (3) (1976) 581–592.
[37] R. Vedantam, I. Fischer, J. Huang, K. Murphy, Generative models of visually grounded imagination, in: International Conference on Learning Representations (ICLR), Vancouver, Canada, 2018.
[38] C.K. Williams, C. Nash, Autoencoders and probabilistic inference with missing data: an exact solution for the factor analysis case, arXiv:1801.03851(2018).
[39] P. McCullagh, Regression models for ordinal data, J. R. Stat. Soc. Ser. B (Methodological) 42 (2) (1980) 109–142.
[40] S. Suh, S. Choi, Gaussian copula variational autoencoders for mixed data, arXiv:1604.04960(2016).
[41] Y. Burda, R. Grosse, R. Salakhutdinov, Importance weighted autoencoders, in: Proceedings of the 4th International Conference on Learning Representations (ICLR-16), San Juan, Puerto Rico, 2016.
[42] X. Li, Z. Chen, L.K.M. Poon, N.L. Zhang, Learning latent superstructures in variational autoencoders for deep multidimensional clustering, in: International Conference on Learning Representations (ICLR), New Orleans, USA, 2019.
[43] M. Lichman, UCI machine learning repository, 2013. http://archive.ics.uci.edu/ml.
[44] M. Azur, E. Stuart, C. Frangakis, P. Leaf, Multiple imputation by chained equations: what is it and how does it work? Int. J. Methods Psychiatr. Res. 20 (1) (2011) 40–49.
[45] I. Valera, Z. Ghahramani, General table completion using a bayesian nonparametric model, in: Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems (NIPS) 27, 2014, pp. 981–989.

Alfredo Nazábal is a Research Associate in the Alan Turing Institute in London under the Defense & Security Programme. Alfredo obtained his Ph.D. in 2017 in Multimedia and Communications from the University Carlos III in Madrid, Spain. His research focuses on deep generative models and artificial intelligence for data analysis, cleaning and wrangling of heterogeneous datasets.

Pablo M. Olmos was born in Granada, Spain, in 1984. He received the B.Sc./M.Sc. and Ph.D. degrees from the University of Sevilla in 2008 and 2011, respectively, all in telecommunication engineering. He is currently an Associate Professor with the Universidad Carlos III de Madrid. He has held appointments as a Visiting Researcher at Princeton University, École Polytechnique Fédérale de Lausanne, Notre Dame University, École Nationale Supérieure de l'Electronique et de ses Applications, and Nokia-Bell Labs. His research interests range from approximate inference methods for Bayesian machine learning to information theory and digital communications. A detailed CV and list of publications can be accessed at http://www.tsc.uc3m.es/olmos.

Zoubin Ghahramani is Professor of Information Engineering at the University of Cambridge and Chief Scientist at Uber. He is also Deputy Director of the Leverhulme Centre for the Future of Intelligence, and a Fellow of St John's College. He was a founding Cambridge Director of the Alan Turing Institute, the UK's national institute for data science. He has worked and studied at the University of Pennsylvania, MIT, the University of Toronto, the Gatsby Unit at University College London, and Carnegie Mellon University. His research focuses on probabilistic approaches to machine learning and artificial intelligence, and he has published over 250 research papers on these topics. He was co-founder of Geometric Intelligence (now Uber AI Labs) and advises a number of AI and machine learning companies. In 2015, he was elected a Fellow of the Royal Society for his contributions to machine learning.

Isabel Valera is a full Professor at the Department of Computer Science of Saarland University, Saarbrücken (Germany) and an independent research group leader at the MPI for Intelligent Systems in Tübingen (Germany). Prior to this, she has held a German Humboldt Post-Doctoral Fellowship, and a "Minerva fast track" fellowship from the Max Planck Society. She obtained her PhD in 2014 and MSc degree in 2012 from the University Carlos III in Madrid (Spain). She worked as postdoctoral researcher at the MPI for Software Systems (Germany) and at the University of Cambridge (UK). Isabel's research focuses on developing machine learning methods that are flexible, robust, interpretable and fair.