Variational Auto-Decoder

Amir Zadeh
LTI, School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
abagherz@cs.cmu.edu

Yao-Chong Lim
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
yaochonl@cs.cmu.edu

Paul Pu Liang
MLD, School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
pliang@cs.cmu.edu

Louis-Philippe Morency
LTI, School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
morency@cs.cmu.edu

Abstract

Auto-Encoding Variational Bayes (AEVB) is one of the most successful algorithms for learning generative models. Approximate posterior inference is an essential step of AEVB, and is commonly done using encoders in a Variational Auto-Encoder (VAE) framework. In many machine learning scenarios, missing data is widespread; hence only incomplete data is observable. A particular question when dealing with incomplete data is: *Is encoder-based posterior approximation reliable in the presence of uncertainties arising from missing information?* If the answer is no, then conditioning the approximate posterior directly on the volatile input may cause major problems for generative modeling, as studied in this paper. When dealing with incomplete data, we show that posterior inference is better done with a proposed encoder-less implementation of AEVB - called Variational Auto-Decoder (VAD) in this paper to distinguish from VAE. Specifically, we show that encoder-based posterior approximation exhibits progressive failure as the amount of missing information increases. VAD, on the other hand, does not suffer from input volatility and shows superior performance in approximate posterior inference.

1 Introduction

AEVB (Auto-Encoding Variational Bayes) is one of the most widely used algorithms for learning generative models [8]. Approximate posterior inference is an important step within AEVB which allows for sampling from the latent space conditioned on the input. VAE (Variational Auto-Encoder [8]), the most well-known implementation of AEVB, relies on an encoder to perform approximate posterior inference. In machine learning, incomplete data, i.e. data with missing values, is widespread. In the realm of incomplete data, encoder-based approaches such as VAE pose a cyclic dependency with generative modeling: to learn a generative model, an encoder needs to be used, and to use an encoder, the missing dimensions need to be replaced, ideally using the underlying data distribution (i.e. the generative model). Commonly, this chain is broken by sub-optimally imputing the missing values (mostly with zeros even in the most recently proposed VAE imputation models [5]). This in turn causes volatility in the input space of the VAE structure, however, the structure is assumed to be able to handle the volatility and reliably perform posterior approximation.

Within the AEVB algorithm, the approximate posterior parameters need not necessarily be inferred using an encoder. As opposed to using an encoder to infer the parameters of the approximate posterior distribution, we propose to optimize the parameters of the approximate posterior directly

without an encoder. This results in an alternative implementation of AEVB, called Variational Auto-Decoder (VAD). Subsequently, using the reparameterization trick [8] for well-known distributions, the approximate posterior parameters can be learned end-to-end by maximizing the variational lower bound, which is differentiable w.r.t the parameters of the well-known distribution (hence gradient-based approaches can be easily used). Within the AEVB learning framework, we specifically study which of VAD or VAE can maximize the variational lower bound more efficiently and learn a more accurate generative model in the presence of missing data. We study this question over multiple datasets from different domains and the following scenarios: 1) similar train and test-time missingness, as well as 2) test-only missingness and train-only missingness.

2 Related Work

Learning from incomplete data is a fundamental research area in machine learning. Notable related works fall into several categories as denoted below.

In a neural framework, Variational Auto-Encoders have been commonly used for learning from incomplete data [9, 17, 10]. A particular implementation based on Conditional Variational Auto-Encoders (C-VAE) has shown to achieve superior performance over existing methods for learning from incomplete data [5].

Generative Adversarial Networks (GANs) have been used for missing data imputation [19]. Aside from being particularly hard to train [15], VAE approaches have shown to perform better in practice [5]. This implementation of VAE is the baseline we compare to in this paper.

Previously proposed Markov-chain based approaches require computationally heavy sampling time and full data to be observable during training [14, 16, 1]. One appeal of these models is that they can directly maximize the evidence (as opposed to the lower bound), however at a heavy computational cost.

Inpainting approaches exist in computer vision which are particularly engineered for visual tasks and sometimes require similar train and test-time missingness for best performance [11, 18].

Approaches have relied on simple learning techniques such as Gaussian Mixture Models [3], Support Vector Machines [12] or Principle Component Analysis [4]. Such models have fallen short in the recent years due to lacking the necessary complexity to deal with increasingly non-linear nature of many real-world datasets.

3 Model

(AEVB) Auto-Encoding Variational Bayes [8] is among the most successful methods for learning generative models. Using a reparameterization trick on a set of known distributions, AEVB allows the learning to be done using SVI (Stochastic Variational Inference [14]). A particularly important step within AEVB is learning an approximate posterior distribution. This approximate posterior is commonly parameterized by a neural network in a VAE (Variational Auto-Encoder [8]). The encoder essentially outputs the parameteres of the approximate posterior.

In this section, we outline an alternative implementation of the AEVB algorithm for the case of incomplete data. We call this approach Variational Auto-Decoder (VAD) since it does not utilize an encoder to infer the paramaters of the approximate posterior. VAD initializes the parameters of the approximate posterior randomly and updates those parameters during the training process using gradient-based methods. We first outline the problem formulation, and subsequently outline the training and inference procedure for VAD.

3.1 Problem Formulation

We assume a ground-truth random variable $\hat{x} \sim p(\hat{x});\ \hat{x} \in \mathbb{R}^d$, sampled from a ground-truth distribution, with d being the dimension of the input space. Unfortunately, the space of $\hat{x}$ is considered to not be fully observable. The part that is observable we denote via random variable x, regarded hereon as *incomplete input*. We assume that a random variable $\alpha \sim p(\alpha); \alpha \in \{0, 1\}^d$ denotes whether or not the data is observable through an indicator in each dimension with value 1 being observable and 0 being missing.

We formalize the process of generating the random variable x as the process of first drawing a ground-truth data sample from $p(\hat{x})$ and a missingness pattern sample from $p(\alpha)$, and subsequently removing information from $\hat{x}$ using α. We draw N i.i.d. samples from the above process to build a dataset.[1] For the rest of this paper, the incomplete dataset is regarded as $X = \{x_1, \ldots, x_N\}$ and the missingness patterns are regarded as $A = \{\alpha_1, \ldots, \alpha_N\}$. The ground-truth dataset is regarded as $\hat{X} = \{\hat{x}_1, \ldots, \hat{x}_N\}$. $\hat{X}$ can never directly be a part of training, validation or testing since it is considered strictly unknown.

3.2 Training

Assuming that data distribution $p(x)$ can be approximated using a parametric family of distributions with the parameters θ, learning can be done by maximizing the likelihood $p(X;\theta)$, w.r.t θ. In practice, the log of the likelihood is often calculated and used. In a latent variable-modeling framework the evidence can often be defined by marginalizing a latent variable as follows:

$$\mathcal{L}(\theta|X) \overset{\text{i.i.d}}{=} \sum_{i=1}^{N} \ln p(x_i;\theta) = \sum_{i=1}^{N} \ln \int p(z, x_i;\theta)\, dz \tag{1}$$

In practice calculating the marginal integral over $p(z, x_i;\theta)$ is either expensive or intractable. Subsequently direct latent posterior inference using $p(z|x_i;\theta)$, which is an essential step in latent variable modeling, becomes impractical.

For any given x_i and any conditional density $q(z|x_i;\phi)$ with z as an unobserved random variable and ϕ as parameters of q, we can rewrite the evidence in Equation 1 as follows:

$$\begin{aligned}\mathcal{L}(\theta; x_i) = &- \int q(z|x_i;\phi) \ln \frac{p(z|x_i;\theta)}{q(z|x_i;\phi)}\, dz \\ &+ \int q(z|x_i;\phi) \ln \frac{p(z|x_i;\theta)p(x_i;\theta)}{q(z|x_i;\phi)}\, dz\end{aligned} \tag{2}$$

With the condition that $q(z|x_i,\phi) > 0$ if $p(z|x_i,\theta) > 0$. To simplify notation, we refer to true posterior $p(z|x_i;\theta)$ as $p_\theta(z|x_i)$ and approximate posterior $q(z|x_i;\phi)$ as $q_\phi(z|x_i)$. More simply, the likelihood in Equation 2 can be written as:

$$\mathcal{L}(\theta|x_i) = \mathrm{KL}\Big(q_\phi(z|x_i) \,\Big|\Big|\, p_\theta(z|x_i)\Big) + \mathcal{V}(q_\phi, \theta|x_i) \tag{3}$$

In the above equation, $\mathrm{KL}(\cdot\,||\,\cdot)$ is the Kullback-Leibler divergence. One can directly minimize this asymmetric divergence and approximate the true posterior using an approximate posterior $q_\phi(\cdot)$. However, doing so requires samples to be drawn from the true posterior. Markov Chain Monte Carlo (MCMC) approaches can be used to draw samples from the true posterior, however, such approaches are usually very costly.

$\mathcal{V}(\cdot)$ is referred to as the Evidence Lower Bound (ELBo) or simply variational lower-bound. It is equal to sum of the expected value of the log of the posterior $p_\theta(z|x_i)$ under distribution $q_\phi(z|x_i)$ and entropy of $q_\phi(z|x_i)$:

$$\mathcal{V}(q_\phi, \theta|x_i) = \mathbb{E}_{q_\phi(z|x_i)}\Big[\ln p_\theta(z, x_i)\Big] + \mathcal{H}\Big(q_\phi(z|x_i)\Big) \tag{4}$$

Through the above formulation, rather than employing a method for learning model parameters through likelihood of data, variational Bayes methods approximate the posterior probability $p_\theta(z|x_i)$ with a simpler distribution $q_\phi(z|x_i)$. Equation 4 can be rewritten as:

$$\mathcal{V}(q_\phi, \theta|x_i) = \mathbb{E}_{q_\phi(z|x_i)}\Big[\ln p_\theta(x|z)\Big] - \mathrm{KL}\Big(q_\phi(z) \,\Big|\Big|\, p_\theta(z)\Big) \tag{5}$$

In the above equation, the first term encourages the latent samples to show high expected likelihood (through reconstruction of x_i) under the approximate posterior distribution, and the second term encourages the latent samples to simultaneously follow the latent prior $p_\theta(z)$.

[1] Notably, each datapoint can have a distinct missingness pattern.

Algorithm 1 Training (and inference) process for the VAD models with multivariate normal distribution as approximate posterior.

$\mathcal{F} : \{\theta^{(0)}\} \leftarrow$ Initialization ▷ Only for training, gets θ^* during inference
$q : \{\mu_i^{(0)}, \Sigma_i^{(0)}\} \leftarrow$ Initialization
repeat:
  $[z] \sim q(z; \mu_i^{(t)}, \Sigma_i^{(t)})$ ▷ Sampling approximate posterior, Equation 6
  $[p(x|z; \theta^{(t)})] = \mathcal{N}(\mathcal{F}([z], \theta^{(t)}); x_i, \Lambda_i)$ ▷ Equation 7
  $\{\theta, \mu_i, \Sigma_i\}^{(t+1)} \leftarrow \underset{\theta, \mu_i, \Sigma_i}{\text{grad\_step}}\Big(\mathcal{V}\big(q^{(t)}, \theta^{(t)}|x_i\big)\Big)$ ▷ No grad_step w.r.t θ during inference
  $t \leftarrow t + 1$
until maximization convergence on $\mathcal{V}$

In a Variational Auto-Decoder framework, the approximate posterior distribution is not parameterized by a neural network, but rather using a well-known distribution directly. Therefore, as opposed to randomly initializing the weights of the encoder, we simply randomly initialize the intrinsic parameters of the approximate posterior. We focus on the family of multivariate Gaussian distributions for this purpose, however other distributions can also be used, as long as reparameterization trick [8] can be defined for them. We define a multivariate Gaussian approximate posterior as:

$$q_\phi(z|x_i) \coloneqq \mathcal{N}(z; \mu_i, \Sigma_i) \tag{6}$$

Note $\phi = \{\mu_i, \Sigma_i\}$ are learnable parameters of this approximate posterior distribution. The reparameterization of this posterior is essentially defined as $z = \mu_i + \epsilon \cdot \Sigma_i$ with $\epsilon \sim \mathcal{N}(0, I)$. Using this reparameterization, the gradient of the lower bound $\mathcal{V}(\cdot)$ can be directly backpropagated to the mean μ_i and variance Σ_i.

Likelihood is similarly defined as a multivariate Gaussian with missing dimensions of x_i marginalized:

$$p(x|z, \Lambda_i) = \mathcal{N}\left(\mathcal{F}(z; \theta); x_i, \Lambda_i\right) \tag{7}$$

This density is centered around x_i as its mean. The covariance Λ_i is defined as a diagonal positive semi-definite matrix with α_i on its main diagonal whenever $\alpha_i \neq 0$. $\mathcal{F}(z; \theta)$ is a neural decoder which takes in the samples drawn from posterior in Equation 6. The optimization is subsequently defined within AEVB: first sampling from the approximate posterior to calculate a Monte-Carlo estimate of the lower bound $\mathcal{V}(\cdot)$, and subsequently maximizing w.r.t θ and ϕ (Equations 6 and 7). Algorithm 1 summarizes the training (and also inference in the next section).

3.3 Inference

Typically, once a generative model is learned, it is used to sample data which belong to the underlying learned distribution. Sampling can be done by sampling from the latent space and subsequently using the decoder to generate the data, with no other steps required.

In certain cases a new data point is given and the goal is to samples the posterior. Calculating the evidence in Equation 1 is still infeasible, even after training is done. Therefore, for the new datapoint, the same variational lower bound $\mathcal{V}(\cdot)$ needs to be maximized. Using the same process as during training framework in Equation 5, the parameters of the approximate posterior are initialized randomly and iteratively updated until convergence. Thus, inference is similar to training, except learned parameters (θ^*) of the decoder are not updated during inference. Once $\mathcal{V}(\cdot)$ is maximized, samples of the approximate posterior can be used to generate similar instances as the given datapoint.

4 Experiments

Based on the Equation 5, the variational lower bound $\mathcal{V}(\cdot)$ is dependent on the expectation of the log likelihood $p(x|z)$ under the approximate posterior $q_\phi(z|x_i)$. This term, which relies on the incomplete input x_i, indicates how well samples drawn from the approximate posterior are able to recreate x_i (using likelihood in Equation 7). If due to the parameterization of the approximate posterior, this term cannot be maximized efficiently for incomplete data during training, then maximizing the lower

bound will subsequently be impacted.[2] For both VAD and VAE,[3] we aim to address whether missing data can cause issues for maximizing this expectation. Therefore, we specifically study the lower bound $\mathcal{V}(\cdot)$ with only the first term to compare if any of the two models inherently fall short in presence of missing data.

In the experiments,[4] both models are trained on identical data and maximize the same lower bound $\mathcal{V}(\cdot)$ as depicted in the previous paragraph. The only difference between the two models is therefore parameterization of the approximate posterior distribution: for VAD it is the parameters of the distributions and for VAE it is the weights of the neural networks. A validation set is used to choose the best performing hyperparameter[5] setup exactly based on the lower bound $\mathcal{V}(\cdot)$. Subsequently, the best trained model is used on test data. The ground truth is never used during training, validation or testing (unless required by the experiment, described later in this section). Only for evaluation purposes, after the inference is done on test set, the ground truth is simply *revealed*. To report a measure that is easy to compare, we report the MSE[6] (Mean Squared Error) between the decoded mean of the approximate posterior in the following categories: 1) Incomplete: we report the MSE between the incomplete data (available dimensions) and the output of the decoder. Since the incomplete data is the basis of the likelihood in Equation 7, we expect models to show low MSE for the incomplete data. 2) Missing: once the inference is done over the incomplete data, missing values are revealed to evaluate the imputation performance of both models. 3) Full: after revealing missing values, we can simply calculate the performance over the full ground-truth data.

Specifically, the following two scenarios are studied in the form of experiments in this paper:

Experiment 1: We study the case where during train and test time, data follows similar a missing rate. Essentially the distribution of missingness is also the same for these cases.

Experiment 2: In real-world situations it is very unlikely that the data will follow the same missing rate during train and test time. Therefore, we also compare the two models for different train and test time missing rates. Since there may be many combinations of missing rates for train and test time, we only study the extreme cases: only test-time missingness (train on ground-truth), and only train-time missingness (test on ground-truth).

4.1 Datasets

We experiment with a variety of datasets from different areas within machine learning. To better understand the ranges of MSE for each dataset, we report a baseline obtained by taking the mean of the ground-truth training data as the prediction during test time. This baseline indicates the limit beyond which models are performing worse than just projecting the mean of the ground-truth data regardless of the input[7].

Toy Synthetic Dataset: We study a case of synthetic data where we control the distributional properties of the data. In the generation process, we first acquire a set of independent dimensions randomly sampled from 5 univariate distributions with uniform random parameters: {`Normal, Uniform, Beta, Logistic, Gumbel`}. Often in realistic scenarios there are inter-dependencies among the dimensions. Hence we proceed to generate interdependent dimensions by picking random subsets of the independent components and combining them using random operations such as weighted multiplication, affine addition, and activation. Using this method, we generate a dataset containing

[2] In simple terms, regardless of the second term in Equation 5, if the approximate posterior and decoder cannot reproduce the data efficiently in the best case, then generative modeling will not be successful, regardless of the second term of Equation 5 (which is anyways the same for both models).

[3] The implementation of VAE in this paper using Equation 7 and with missing mask is identical to a model which was published during preparation of this paper, called VAEAC by [5]. Missing values replaced by zeros before inputting to encoder. The authors did substantial experiments and found that this model is comparable or better than previous approaches in most data imputation and posterior approximation tasks studied in their paper.

[4] Code and data available through `https://github.com/A2Zadeh/Variational-Autodecoder`

[5] Both models undergo substantial hyperparameter search as described in Appendix A (with exact values). Hyperparameters include (but not limited to) the number of layers in the decoder (and encoder for VAE), the number of neuron in each layer, and the latent space dimensions.

[6] MSE is calculated per each dimension, therefore it is independent of the missing rate.

[7] With a very minimal deviation across experiments for each dataset, this threshold also applies to missing and incomplete components.

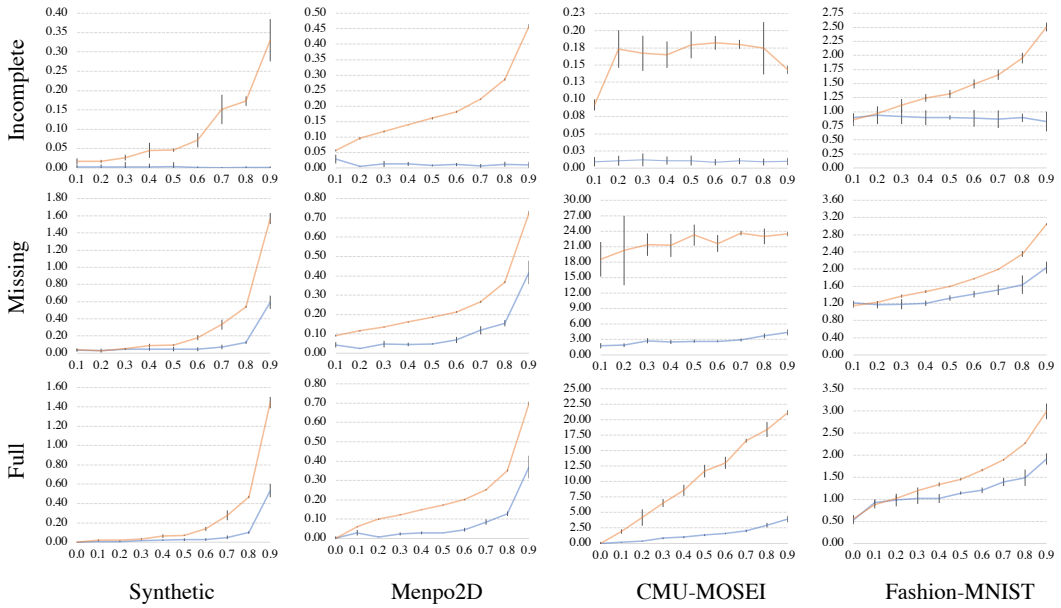

Figure 1: Best viewed zoomed in and in color. The results of the Experiment 1 (Section 4.2) for incomplete, missing and full categories. Blue curve shows VAD results and orange curve shows VAE results. The x axis denotes the missing rate r and the y axis is the reconstruction Mean Squared Error (MSE, lowe better). The standard deviation is calculated based on 5 test runs of the best performing model on the validation set. The gap between both models becomes larger as the missing ratio r increases. In the full category, $r = 0.0$ shows the performance of the case where there is no missing data (train or test). The performance thresholds from left to right are: 5.46, 0.62, 50.83, 8.64 indicating the MSE beyond which models are predicting worse than average of each dimension regardless of input.

$N = 50,000$ datapoints with ground-truth dimension $d = 300$. Further details of the generation and exact ranges are given in the Supplementary Material. The threshold MSE for this dataset is 5.46 on this dataset.

Menpo Facial Landmark Dataset: Menpo2D contains $13,391$ facial images with various subjects, expressions, and poses [21]. Due to these variations, the nature of the dataset is complex. Since Menpo dataset has ground truth annotations for 84 landmarks regardless of self-occlusions in the natural image, it allows for creating a real-life ground-truth data for our experiments. The purpose of using this dataset is to compare the two models on how well they recreate the structure of an object given only a subset of available keypoints. The threshold MSE for this dataset is 0.62.

CMU-MOSEI Dataset: CMU Multimodal Sentiment and Emotion Intensity (CMU-MOSEI) is an in-the-wild dataset of multimodal sentiment and emotion recognition [20]. The datasets consist of sentences utterance from online YouTube monologue videos. CMU-MOSEI consists of 23,500 such sentences and with three modalities of text (words), vision (gestures) and acoustic (sound). For text modality, the datasets contains GloVe word embeddings [13]. For visual modality, the datasets contains facial action units, facial landmarks, and head pose information. For acoustic modality, the datasets contain high and low-level descriptors following COVAREP [2]. We use expected multimodal context for each sentence, similar to unordered compositional approaches in NLP [6]. The threshold MSE for this dataset is 50.83.

Fashion-MNIST: Fashion-MNIST[8] is a variant of the MNIST dataset. It is considered to be more challenging than MNIST since variations within fashion items are usually more complex than written digits. The dataset consists of $70,000$ grayscale images with shape 28×28 from 10 fashion items. The threshold MSE is 8.64 for this dataset.

We base our experiments on Missing Completely at Random (MCAR), which is a severe case of missingness. For each $\hat{x}_i$, we sample a missing pattern $\alpha_i \sim \texttt{Bernouli}(r); \alpha_i \in \{0, 1\}^d$ with missing ratio r ranging from 0.1 to 0.9 with increments of 0.1. This form of α_i essentially allows each dimension to unexpectedly go missing.

[8] https://github.com/zalandoresearch/fashion-mnist

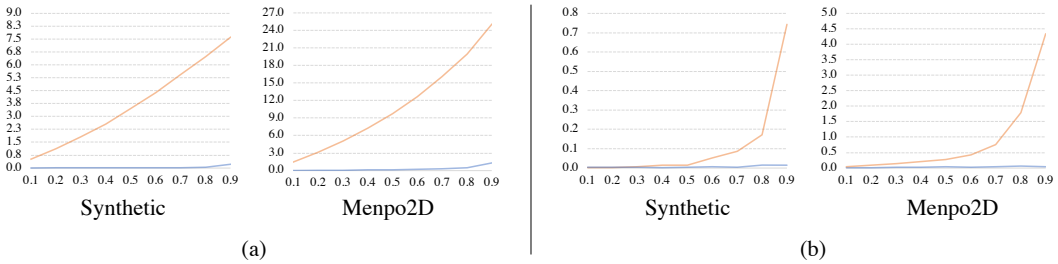

Figure 2: Best viewed zoomed in and in color. Results (in Full cateogry) of Experiment 2 (Section 4.3) for (a) test-time missingness and (b) train-time missingness. Blue curve shows VAD results and orange curve shows VAE results. The x axis denotes the missing rate r and the y axis is the reconstruction Mean Squared Error (MSE). In both the experiments VAD shows superior performance than VAE. The performance of VAE is significantly affected in both scenarios.

4.2 Experiment 1

In this experiment, both train and test data follow the same missing ratio r. For each r, models are trained using likelihood in Equation 7 and maximize the lower-bound $\mathcal{V}(\cdot)$ in Equation 5. Figure 1 shows the results of this experiment for the best validated models for both VAD and VAE. In all the three incomplete, missing (imputation) and full (ground-truth) categories, VAD shows superior performance than VAE. As the missingness increases, the gap between the two models widens in all three categories (except for CMU-MOSEI where the performance gap is large in incomplete and missing even for small r). This essentially states that VAE becomes increasingly unstable in presence of missing data. Specifically for the case of incomplete data, VAE is not able to perform reliable posterior inference since the output of the decoder increasingly deviates farther away from the available input. VAD on the other hand, shows steady performance in the incomplete category. The performance of both models is naturally affected for the missing category as r increases (in reality some of the missing data may not be imputable given the available input). However, the increasing gap between the two models also appears in missing category. Finally the comparison in the full category shows that VAD is able to regenerate the ground-truth better than VAE. In a full picture, Figure 1 suggests that approximate posterior using an encoder conditioned on a volatile input becomes increasingly unstable as missingness becomes more severe.

4.3 Experiment 2

While in the previous experiment both the train and test stages followed the same missing rate, realistic scenarios are often more complex. In the most extreme cases, we study two possible scenarios: test-only missingness and train-only missingness. For this experiment, we choose the best performing models from the previous experiment based on their performance on incomplete data in validation set.

In the test-time missingness scenario, models are trained on the ground-truth data without any expectation that during test-time an arbitrary subset of the data may go missing. Essentially, during test-time this assumption proves to be wrong and the data indeed goes missing exactly following a missing ratio r ranging from 0.1 to 0.9. Figure 2 shows the results of this experiment for the synthetic and Menpo2D datasets in Full category.[9] In both cases the performance of VAE is substantially affected by the missing dimensions during test-time, achieving far inferior performance than the case in Experiment 1 (being trained on the same missingness). The performance of VAD remains almost similar for both synthetic and Menpo2D datasets and relatively similar to Experiment 1. We also visually demonstrate this in an inpainting scenario. We compare models when they are trained on ground-truth and tested on incomplete data against when they are trained on similar missing ratio. Both models are trained on the Fashion-MNIST ground-truth train set and subsequently during testing the data may go missing. Figure 3 shows the test-time performance for both models for different missing ratios as well as block-sized missingness. Visually, it can be seen that VAE suffers heavily if it is trained on ground-truth but data goes missing during test-time. In fact, in high missing rate, VAE simply blurs out around the available datapoints while VAD is able to recover the missing areas of

[9]Due to space constraint we only report for 2 datasets in Full category.

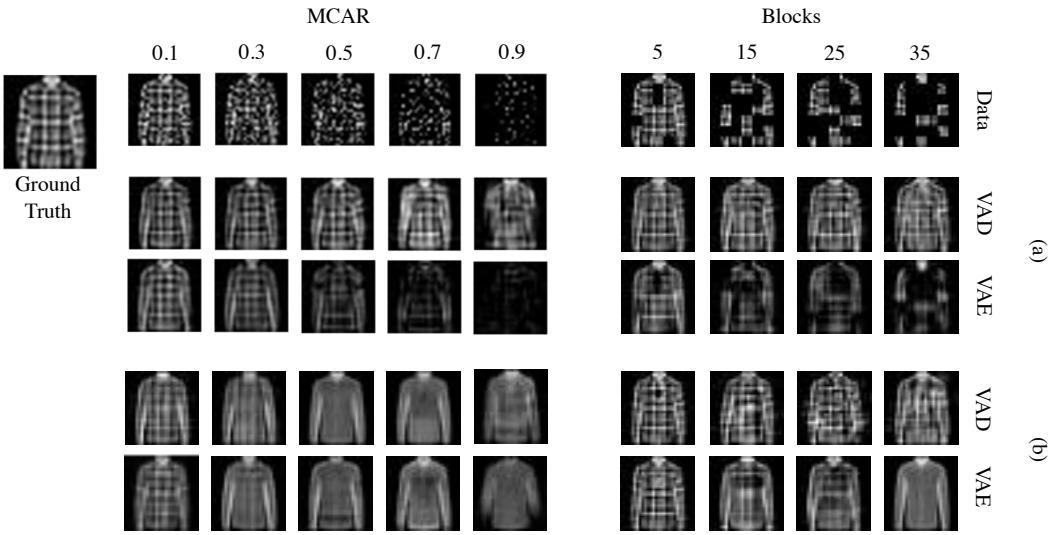

Figure 3: Visualization of inpainting for experiments on Fashion-MNIST. Top row (Data) shows the given data to both VAD and VAE. Ground-truth is the real image from Fashion-MNIST. For (a), training is done on ground-truth data and testing is done on incomplete data. For (b) training and testing is done on similar missing rate. For the case of MCAR (a), VAE significantly deteriorates when trained on ground-truth data and given incomplete data. This trend is also visible but at a much slower rate for the case (b) - where only at $r = 0.9$ VAE shows visually perceivable failure. We also compare the performance of both models when the missingness changes from MCAR random missing 4×4 blocks - training for case (a) still done on ground-truth and (b) done in presence of random blocks. VAD is able to maintain a better performance in both (a) and (b) for random blocks. Overall, (a) shows that VAE mostly focuses on the given areas of the image and mostly reconstructs that area. This is not the case for the VAD, which recreates the other areas of the image as well.

the image. Compared to the case where missing ratio is the same, we observe that it is crucial for VAE to train on the same missing rate as the test time, while VAD does not suffer from this.

The train-time missingness scenario is the opposite of the above scenario.[10] Models are trained on a train set with a missing ratio r ranging from 0.1 to 0.9. During testing, they perform inference on a different missing rate, in the extreme case on ground-truth test. The right side of Figure 2 shows the results of this experiment on both VAD and VAE. We observe a similar trend of performance between the two models, with VAD remaining consistent while the performance of VAE deteriorates as missing rate increases.

5 Conclusion

In this paper, we proposed an alternative implementation of the AEVB for the case of incomplete data, called Variational Auto-Decoder (VAD). We studied the effect of missing data on the approximate posterior conditioned directly using an encoder on the incomplete input (i.e. VAE). We showed that such conditioning may not allow for maximizing the variational lower bound efficiently, due to poor performance for maximizing the expected likelihood (under the approximate posterior) of the incomplete data. We showed that VAD is better suited for this case since it does not take the volatile data as input. The approximate posterior in VAD is parameterized by a known distribution, parameters of which are directly optimized in a variational learning framework. For VAD, similar to VAE, the parameters of the approximate posterior can be learned using gradient-based approaches. When train and test followed similar missing ratios, our experimental results showed superior performance of VAD. This was extended to cases of only test-time missingness and only train-time missingness, where VAD showed superior performance.

[10] Not to be confused with denoising methods or dropout which map noisy input to the ground-truth during train time. In this scenario the ground-truth training set can never be fully observed for training.

References

[1] Florian Bordes, Sina Honari, and Pascal Vincent. Learning to generate samples from noise through infusion training. *arXiv preprint arXiv:1703.06975*, 2017.

[2] Gilles Degottex, John Kane, Thomas Drugman, Tuomo Raitio, and Stefan Scherer. Covarep: A collaborative voice analysis repository for speech technologies. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 960–964. IEEE, 2014.

[3] Olivier Delalleau, Aaron Courville, and Yoshua Bengio. Efficient em training of gaussian mixtures with missing data. *arXiv preprint arXiv:1209.0521*, 2012.

[4] Stéphane Dray and Julie Josse. Principal component analysis with missing values: a comparative survey of methods. *Plant Ecology*, 216(5):657–667, 2015.

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

[6] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1681–1691, 2015.

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[8] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[9] John T McCoy, Steve Kroon, and Lidia Auret. Variational autoencoders for missing data imputation with application to a simulated milling circuit. *IFAC-PapersOnLine*, 51(21):141–146, 2018.

[10] Alfredo Nazabal, Pablo M Olmos, Zoubin Ghahramani, and Isabel Valera. Handling incomplete heterogeneous data using vaes. *arXiv preprint arXiv:1807.03653*, 2018.

[11] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[12] Kristiaan Pelckmans, Jos De Brabanter, Johan AK Suykens, and Bart De Moor. Handling missing values in support vector machine classifiers. *Neural Networks*, 18(5-6):684–692, 2005.

[13] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. 2014.

[14] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[15] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[16] Jascha Sohl-Dickstein, Eric A Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *arXiv preprint arXiv:1503.03585*, 2015.

[17] Christopher KI Williams, Charlie Nash, and Alfredo Nazábal. Autoencoders and probabilistic inference with missing data: An exact solution for the factor analysis case. *arXiv preprint arXiv:1801.03851*, 2018.

[18] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6721–6729, 2017.

[19] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. Gain: Missing data imputation using generative adversarial nets. *arXiv preprint arXiv:1806.02920*, 2018.

[20] AmirAli Bagher Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2236–2246, 2018.

[21] Stefanos Zafeiriou, George Trigeorgis, Grigorios Chrysos, Jiankang Deng, and Jie Shen. The menpo facial landmark localisation challenge: A step towards the solution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, volume 1, page 2, 2017.

Hyperparameter	Group	Values
# of latent variables	A B	25, 50, 100 50, 100, 400
# of hidden units per layer	A B	50, 100, 200 100, 200, 400
# of hidden layers	A, B	2, 4, 6
LR of network parameters and latent variables	A, B	$10^{-2}, 10^{-3}$

Table 1: Hyperparameters used for the experiments on VAD and VAE across different datasets. Group B refers to Fashion-MNIST, while group A refers to all other datasets.

Distribution	Parameter	Range of values
$\texttt{Normal}(\mu, \sigma)$	μ σ	$[-1, 1]$ $(0, 2]$
$\texttt{Uniform}(a, b)$	a b	$[-2, 2]$ $[a, 2]$
$\texttt{Beta}(\alpha, \beta)$	α β	$[0, 3]$ $[0, 3]$
$\texttt{Logistic}(\mu, s)$	μ s	$[-1, 1]$ $(0, 2]$
$\texttt{Gumbel}(\mu, \beta)$	μ β	$[-1, 1]$ $(0, 2]$

Table 2: Parameters used during generation of synthetic data.

Appendices

A Implementation Remarks

Here we detail the hyperparameter space used for the experiments. Where possible, we tried to establish a fair comparison between the VAD and VAE models by using similar hyperparameters, however both models underwent substantial grid search. We varied three main hyperparameters: the dimensionality of the posterior space d_z, the number of feedforward hidden layers in the decoder $\mathcal{F}$ and the encoder (encoder only for VAE), and the number of hidden units in each hidden layer. A summary is shown in Table 1.

During inference of VAD models, we simply stopped once the model reached a plateau. Since VAD models have a high degree of freedom for approximate posterior $q_\phi(\cdot)$, we observed that it is crucial to use Adam [7] for learning the parameters of the approximate distribution. For both models, learning rates of $10^{\{-2,-3\}}$ for the latent variables and hidden units were used.

During inference, both models use an approximate posterior with a learnable variance. However, in practice when using incomplete data, learning the variance was a very unstable process for VAE. VAE models showed very high sensitivity to even small variances, quickly performing similar to projecting the mean in each dimension. We believe combination of noise from imputed values and also the noise added through approximate posterior may have too much cause uncertainties for VAE. While VAD models suffered similarly from the same instability during learning the variance, the performance was better than VAE.

Therefore, during our experiments we treated the approximate posterior variance as a hyperparameter and trained the models for different variances. This way results improved substantially. The best performing variance may change depending on the problem and the range of the input space, however, in general we observed that very large variances did not converge well, while small variances did not yield the best results.

A.1 Synthetic Data Generation

The parameters of the synthetic data are outlined in Table 2.