On casting importance weighted autoencoder to an EM algorithm to learn deep generative models

Donghha Kim
Seoul National University
South Korea

Jaesung Hwang
SK Telecom
South Korea

Yongdai Kim
Seoul National University
South Korea

Abstract

We propose a new and general approach to learn deep generative models. Our approach is based on a new observation that the importance weighted autoencoders (IWAE, Burda et al. [2015]) can be understood as a procedure of estimating the MLE with an EM algorithm. Utilizing this interpretation, we develop a new learning algorithm called importance weighted EM algorithm (IWEM). IWEM is an EM algorithm with *self-normalized* importance sampling (snIS) where the proposal distribution is carefully selected to reduce the variance due to snIS. In addition, we devise an annealing strategy to stabilize the learning algorithm. For missing data problems, we propose a modified IWEM algorithm called miss-IWEM. Using multiple benchmark datasets, we demonstrate empirically that our proposed methods outperform IWAE with significant margins for both fully-observed and missing data cases.

1 Introduction

Probabilistic generative models with deep neural networks have achieved tremendous success for modeling high dimensional data due to the development of the variational autoencoding framework (VAE, [Kingma and Welling, 2013, Rezende et al., 2014]). VAE models the distribution of an observable random vector $\mathbf{x}$ of high dimension by introducing a lower dimensional latent vector $\mathbf{z}$ such that $p(\mathbf{x};\theta) = \int_{\mathbf{z}} p(\mathbf{x}|\mathbf{z};\theta)p(\mathbf{z};\theta)d\mathbf{z}$, where θ is the parameter of the model. Instead of maximizing the marginal log-likelihood which is computationally infeasible, VAE utilizes the lower bound of the marginal log-likelihood, called ELBO, which is more tractable to compute.

Various more tight but still tractable lower bounds than ELBO have been proposed by Burda et al. [2015], Cremer et al. [2017], Kingma et al. [2016], Rezende and Mohamed [2015], Salimans et al. [2015], Sø nderby et al. [2016]. Especially, the importance weighted autoencoders (IWAE, Burda et al. [2015]) use multiple samples from the variational distribution to construct a lower bound, which improves VAE significantly.

In this paper, we propose a new learning algorithm for probabilistic generative models based on the expectation-maximization (EM) algorithm which tries to maximize the marginal log-likelihood directly instead of pursuing a tighter lower bound. The proposed learning algorithm is motivated by uncovering the relation between IWAE and the EM algorithm. We first show that IWAE can be understood as a version of the (generalized) EM algorithm with *self-normalized* importance sampling (snIS). That is, in fact IWAE tries to estimate the maximum likelihood estimate (MLE) directly. This new perspective explains partly the superiority of IWAE since learning methods based on ELBO have suffered from sub-optimality of their estimates due to inevitable discrepancy between the model posterior and the variational distribution.

Based on this new interpretation of IWAE, we propose an EM algorithm called importance weighted EM (IWEM) algorithm[1], which improves IWAE. The main idea of IWEM is to use snIS to approximate the E-step with a carefully selected proposal distribution to reduce the variance raised by snIS. In addition, we devise an annealing strategy to stabilize the EM algorithm in early learning phases.

An appealing feature of IWEM is that it can be modified easily for a nonstandard case. For example, IWEM can be applied to missing data problems by modifying the selection of the proposal distribution in snIS

Proceedings of the 23[rd]International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

[1]Code available at `https://github.com/dongha0718/IWEM`

slightly.

By conducting test log-likelihood comparisons for multiple benchmark datasets, we show empirically that IWEM outperforms IWAE with large margins, and is also superior to other recent methods. Particularly for missing data scenarios, we observe that the margin between IWEM and IWAE becomes larger as the missing rate increases.

There are some similar methods to IWEM. MCEM [Song et al., 2016] is a method of using Monte Carlo method in E-step as IWEM does. While we focus to E-step and P-step by reducing variance due to snIS, MCEM mainly modified M-step by introducing auxiliary random variables. Dieng and Paisley [2019] have also improved IWAE by the idea of re-interpreting IWAE as an EM algorithm with snIS. They exploited the inclusive KL divergence and hyperproposal in order to induce a good proposal. In contrast, we stick to use the same divergence as the one used in IWAE. Moreover, we provide a theoretical justification of using the divergence in IWAE

The paper is organized as follows. In Section 2, we provide brief explanations of related works, and the detailed descriptions of our methods are given in Section 3. Application of IWEM for missing data case is described in Section 4 and results of numerical experiments including test log-likelihood performance analysis are presented in Section 5. Lastly final conclusions follow in Section 6.

2 Related Work

2.1 EM algorithm

The EM algorithm is an efficient iterative method to compute the MLE in the presence of missing data or latent random variable. Let $\mathbf{x}$ be an observable random vector and $\mathbf{z}$ be a missing or latent random vector. To maximize the marginal log-likelihood function $\log p(\mathbf{x};\theta)$ which is parametrized by θ, EM algorithm alternates the following two steps iteratively: E-step and M-step.

Let $\theta^{(t)}$ be the current estimate of θ at iteration t. In E-step, we define $Q(\theta|\theta^{(t)};\mathbf{x})$ as the expected value of the joint log-likelihood function $p(\mathbf{x},\mathbf{z};\theta)$ with respect to the current conditional distribution $p(\mathbf{z}|\mathbf{x};\theta^{(t)})$ given as

$$Q(\theta|\theta^{(t)};\mathbf{x}) := \int p(\mathbf{z}|\mathbf{x};\theta^{(t)}) \cdot \log p(\mathbf{x},\mathbf{z};\theta) d\mathbf{z}. \qquad (1)$$

In M-step, we update the current estimate $\theta^{(t+1)}$ by maximizing $Q(\theta|\theta^{(t)};\mathbf{x})$ with respect to θ. It is also possible to choose $\theta^{(t+1)}$ which simply increases $Q(\theta|\theta^{(t)};\mathbf{x})$ so that $Q(\theta^{(t+1)}|\theta^{(t)};\mathbf{x}) \geq Q(\theta^{(t)}|\theta^{(t)};\mathbf{x})$. This kind of the modified algorithm is called a generalized EM algorithm. Hereafter we use the two terms "EM algorithm" and "generalized EM algorithm" interchangeably if there is no confusion.

When it is intractable to calculate (1), one may use snIS to approximate (1) by introducing a proposal distribution $q(\mathbf{z}|\mathbf{x};\phi)$ parametrized by ϕ, which is given as

$$\widehat{Q}(\theta|\theta^{(t)},\phi;\mathbf{x}) := \sum_{k=1}^{K} \frac{w_k}{\sum_{k'} w_{k'}} \log p(\mathbf{x},\mathbf{z}_k;\theta), \qquad (2)$$

where $\mathbf{z}_k \sim q(\mathbf{z}|\mathbf{x};\phi)$ and $w_k = p(\mathbf{x},\mathbf{z}_k;\theta^{(t)})/q(\mathbf{z}_k|\mathbf{x};\phi)$ for $k = 1,...,K$. If necessary, one also updates ϕ to encourage $q(\mathbf{z}|\mathbf{x};\phi)$ to be a good proposal distribution. In this study we call this procedure the proposal step (P-step).

The EM algorithm has not been used popularly for learning probabilistic generative models since the choice of the proposal distribution is not easy and thus variance in IS (or snIS) is rather large [Bengtsson et al., 2008, Dowling et al., 2018, Tokdar and Kass, 2010]. In this paper, we propose an efficient way of selecting the proposal distribution in P-step.

2.2 Variational autoencoders

Variational autoencoder (VAE, [Kingma and Welling, 2013, Rezende et al., 2014]) maximizes the lower bound of the marginal log-likelihood, called evidence lower bound (ELBO) with respect to a variational distribution $q(\mathbf{z}|\mathbf{x};\phi)$ parametrized by ϕ given as

$$\begin{aligned} \log p(\mathbf{x};\theta) &= \log \int q(\mathbf{z}|\mathbf{x};\phi) \cdot \frac{p(\mathbf{x},\mathbf{z};\theta)}{q(\mathbf{z}|\mathbf{x};\phi)} d\mathbf{z} \\ &\geq \int q(\mathbf{z}|\mathbf{x};\phi) \cdot \log \left[\frac{p(\mathbf{x},\mathbf{z};\theta)}{q(\mathbf{z}|\mathbf{x};\phi)} \right] d\mathbf{z} \\ &=: L^{\mathrm{VAE}}(\theta,\phi;\mathbf{x}). \end{aligned}$$

In practice VAE approximates ELBO by using the Monte Carlo method,

$$\widehat{L}^{\mathrm{VAE}}(\theta,\phi;\mathbf{x}) := \frac{1}{L} \sum_{l=1}^{L} \log \left(\frac{p(\mathbf{x},\mathbf{z}_l;\theta)}{q(\mathbf{z}_l|\mathbf{x};\phi)} \right), \qquad (3)$$

where $\mathbf{z}_l \sim q(\mathbf{z}|\mathbf{x};\phi)$ for $l = 1,...,L$, and maximizes (3) with respect to θ and ϕ.

2.3 Importance weighted autoencoders

Importance weighted autoencoder (IWAE) of Burda et al. [2015] is a variational inference strategy capable of producing arbitrarily tight lower bounds. For a given

Algorithm 1: IWAE as EM algorithm

Require: Train dataset: $\mathcal{D} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$
Require: Model architectures: $p(\mathbf{x}, \mathbf{z}; \theta)$ and $q(\mathbf{z}|\mathbf{x}; \phi)$
Require: Initial parameters: $\theta^{(1)}$ and $\phi^{(1)}$
Require: SGD based optimization algorithm: $\mathcal{L}(loss, params, current\_params)$
Initialization: Parameters $\theta^{(1)}$ and $\phi^{(1)}$
Initialization: t $\leftarrow 1$
while $\theta^{(t)}$ "not converge" **do**
  E-step Calculate $\sum_{i=1}^{n} \widehat{L}^{\text{IWAE}}(\theta, \phi^{(t)}; \mathbf{x}_i)$ in (4).
  M-step Update $\theta^{(t+1)}$:

$$\theta^{(t+1)} \leftarrow \mathcal{L}\left(-\sum_{i=1}^{n} \widehat{L}^{\text{IWAE}}(\theta, \phi^{(t)}; \mathbf{x}_i), \theta, \theta^{(t)}\right).$$

  P-step Update $\phi^{(t+1)}$:

$$\phi^{(t+1)} \leftarrow \mathcal{L}\left(-\sum_{i=1}^{n} \widehat{L}^{\text{IWAE}}(\theta^{(t+1)}, \phi; \mathbf{x}_i), \phi, \phi^{(t)}\right).$$

end

$\mathbf{x}$, IWAE maximizes the following function

$$\mathbb{E}_{\mathbf{z}_1,...,\mathbf{z}_K \sim q(\mathbf{z}|\mathbf{x};\phi)}\left[\log\left(\frac{1}{K}\sum_{k=1}^{K}\frac{p(\mathbf{x}, \mathbf{z}_k; \theta)}{q(\mathbf{z}_k|\mathbf{x}; \phi)}\right)\right],$$

which uses multiple samples (i.e. K many samples) from the variational distribution $q(\mathbf{z}|\mathbf{x}; \phi)$. In practice, IWAE uses an approximated version of the above lower bound by utilizing the Monte Carlo method which is given as

$$\widehat{L}^{\text{IWAE}}(\theta, \phi; \mathbf{x}) := \log\left(\frac{1}{K}\sum_{k=1}^{K}\frac{p(\mathbf{x}, \mathbf{z}_k; \theta)}{q(\mathbf{z}_k|\mathbf{x}; \phi)}\right), \qquad (4)$$

where $\mathbf{z}_k \sim q(\mathbf{z}|\mathbf{x}; \phi)$ for $k = 1, ..., K$.

3 Proposed method

3.1 IWAE as EM algorithm

As mentioned in Section 1, our method is motivated by close investigation of relation between IWAE and the EM algorithm. The following proposition is a key result to interpret IWAE as an EM algorithm whose proof is in the supplementary material.

Proposition 1 *The following equality holds for any θ':*

$$\nabla_\theta \widehat{L}^{IWAE}(\theta, \phi; \mathbf{x})\Big|_{\theta=\theta'} = \nabla_\theta \widehat{Q}(\theta|\theta', \phi; \mathbf{x})\Big|_{\theta=\theta'}.$$

Proposition 1 implies that IWAE is equivalent to the EM algorithm for learning θ if we use a gradient based optimization algorithm. The step of updating ϕ in IWAE can be understood as P-step. Consequently, we can conclude that IWAE is a method trying to find the MLE directly. We summarize this new interpretation of IWAE in Algorithm 1.

Conceptually, for given θ, P-step is to find ϕ such that $q(\mathbf{z}|\mathbf{x}; \phi)$ is as close to $p(\mathbf{z}|\mathbf{x}; \theta)$ as possible. Note that the objective function of P-step in Algorithm 1 is equivalent to

$$-\log\left(\frac{1}{K}\sum_{k=1}^{K}\frac{p(\mathbf{z}_k|\mathbf{x}; \theta^{(t+1)})}{q(\mathbf{z}_k|\mathbf{x}; \phi)}\right) \qquad (5)$$

since $p(\mathbf{x}; \theta^{(t+1)})$ is irrelevant to ϕ, and (5) is an unbiased estimate of

$$\mathbb{E}_{\mathbf{z}_1,...,\mathbf{z}_K \sim q(\mathbf{z}|\mathbf{x};\phi)}\left[-\log\left(\frac{1}{K}\sum_{k=1}^{K}\frac{p(\mathbf{z}_k|\mathbf{x}; \theta^{(t+1)})}{q(\mathbf{z}_k|\mathbf{x}; \phi)}\right)\right]. \qquad (6)$$

Thus we can say that IWAE encourages the proposal distribution $q(\mathbf{z}|\mathbf{x}; \phi)$ to be similar to the current model posterior $p(\mathbf{z}|\mathbf{x}; \theta^{(t+1)})$ and the similarity is measured by (6). When $K = 1$, (6) becomes the standard Kullback-Leibler (KL) divergence. When $K > 1$, however, (6) seems to be a new but interesting divergence. Of course (6) is minimized when $q(\mathbf{z}|\mathbf{x}; \phi) = p(\mathbf{z}|\mathbf{x}; \theta^{(t+1)})$.

3.2 Theoretical analysis of (6)

Dieng and Paisley [2019] pointed out that (6) does not correspond to minimizing any divergence, leading to the poor proposal distribution. Here, we provide a theoretical justification of using (6) as a divergence.

Proposition 2 *For any two density functions $p(\mathbf{z})$ and $q(\mathbf{z})$, consider the new divergence*

$$D^{IW}(q||p) := -\mathbb{E}_{\mathbf{z}_1,...,\mathbf{z}_K \sim q}\left[\log\left(\frac{1}{K}\sum_{k=1}^{K}\frac{p(\mathbf{z}_k)}{q(\mathbf{z}_k)}\right)\right].$$

If p/q is bounded, then we have

$$\lim_{K\to\infty} 2K \cdot D^{IW}(q||p) = \chi^2(p||q),$$

where $\chi^2(p\|q)$ is the Chi-squared distance between p and q.

The proof is in the supplementary material. Proposition 2 implies that minimizing (6) enforces the proposal distribution to be similar to the model posterior in the sense of the Chi-square distance.

3.3 IWEM

In this section we propose IWEM which adds two new techniques to the interpretation of IWAE as an EM algorithm.

3.3.1 Optimal P-step

Using $\widehat{Q}(\theta|\theta^{(t)}, \phi^{(t)}; \mathbf{x})$ in (2) as the objective function instead of $Q(\theta|\theta^{(t)}; \mathbf{x})$ in (1) inevitably causes additional variance. Thus we need to find a good proposal distribution which yields the additional variance as small as possible.

It is well known [Owen, 2013] that the optimal proposal distribution, which minimizes the variance due to IS, is given by the following form

$$\begin{aligned} q^{\text{opt}}(\mathbf{z}) &\propto \left|\log p(\mathbf{x}, \mathbf{z}; \theta^{(t+1)})\right| \cdot p(\mathbf{x}, \mathbf{z}; \theta^{(t+1)}) \\ &=: \tilde{p}(\mathbf{z}|\mathbf{x}; \theta^{(t+1)}). \end{aligned}$$

Recall that IWAE finds the proposal distribution similar to $p(\mathbf{z}|\mathbf{x}; \theta^{(t+1)}) \propto p(\mathbf{x}, \mathbf{z}; \theta^{(t+1)})$, which means that P-step of IWAE may not be optimal for the EM algorithm with IS.

By adopting the results of Owen [2013] and Proposition 2, we propose a new P-step by replacing $\widehat{L}^{\text{IWAE}}$ to

$$\widehat{L}^{\text{opt}}(\theta^{(t+1)}, \phi; \mathbf{x}) := \log \left(\frac{1}{J} \sum_{j=1}^{J} \frac{\tilde{p}(\mathbf{z}_j|\mathbf{x}; \theta^{(t+1)})}{q(\mathbf{z}_j|\mathbf{x}; \phi)} \right), \quad (7)$$

where J is the number of samples from $q(\mathbf{z}|\mathbf{x}; \theta^{(t+1)})$. We call this modification *optimal P-step*. It helps the proposal distribution to be similar to $q^{\text{opt}}(\mathbf{z})$ and does leads to yield small variance due to IS. The effectiveness of *optimal P-step* is illustrated in Figure 1.

Remark Since we use snIS, not the vanilla IS, the optimal proposal density [Owen, 2013] should be

$$\begin{aligned} q^{\text{opt}}(\mathbf{z}) \propto & \left|\log p(\mathbf{x}, \mathbf{z}; \theta^{(t+1)}) - Q(\theta^{(t+1)}|\theta^{(t+1)}, \phi; \mathbf{x})\right| \\ & \times p(\mathbf{x}, \mathbf{z}; \theta^{(t+1)}). \end{aligned}$$

But it would not be possible to calculate the term Q exactly, and therefore we use (7) as the objective function in P-step.

3.3.2 Annealing strategy

The idea is motivated from the comparison of variances between (2) and (3). Note that ELBO in (3) is equivalent to

$$\frac{1}{L} \sum_{l=1}^{L} \log p(\mathbf{x}, \mathbf{z}_l; \theta),$$

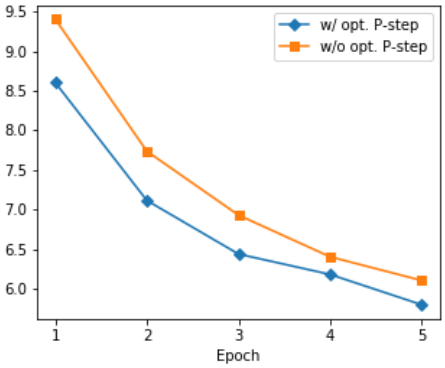

Figure 1: Variance of (2) with (**blue**) and without (**orange**) *optimal P-step*. For each method, we calculate 100 many values of (2) by applying 100 IS at each training epochs of the static biMNIST dataset, and obtain the corresponding variance. The values in the plot are the averages of the variances of (2) for all the train samples.

when we update θ. We can easily notice that, to approximate each expected joint log-likelihood, (2) utilizes the snIS while (3) utilizes the Monte Carlo method.

At initial learning stage there is a large discrepancy between $p(\mathbf{z}|\mathbf{x}; \theta)$ and $q(\mathbf{z}|\mathbf{x}; \phi)$. It leads to large variance of (2), which may hamper the learning procedure. However this discrepancy does not affect the variance of (3), which is empirically illustrated in Figure 2.

Based on the above observation, we propose to apply the idea of *warm-up* [Bowman et al., 2015] to IWEM in order to reduce the variance at early learning stages. We devise a technique, called *annealing strategy* to modify the E-step by taking a convex combination with (2) and (3) which is formulated as

$$\begin{aligned} \widehat{Q}^{\alpha}(\theta|\theta^{(t)}, \phi^{(t)}; \mathbf{x}) := & \ \alpha \cdot \widehat{Q}(\theta|\theta^{(t)}, \phi^{(t)}; \mathbf{x}) \\ & + (1-\alpha) \cdot \widehat{L}^{\text{VAE}}(\theta, \phi^{(t)}; \mathbf{x}), \end{aligned} \quad (8)$$

where $\alpha \in [0, 1]$ is called *annealing controller*. We assign α to zero in the initial stage and increase it incrementally up to one as the learning iteration proceeds. *Annealing strategy* reduces the variance of the objective function in E-step at early learning stages and thus stabilizes the overall learning procedure.

The algorithm of IWEM, which uses *optimal P-step* and *annealing strategy*, is summarized in Algorithm 2.

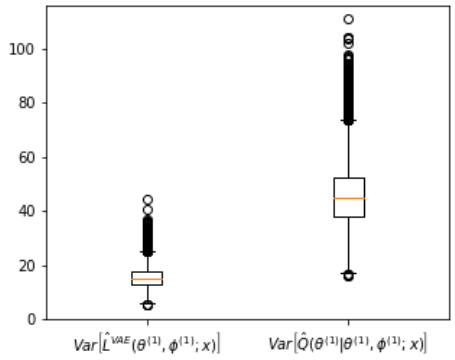

Figure 2: Boxplots of variances of the approximated expected joint log-likelihoods (2) and (3) for VAE and EM, respectively on the static BiMNIST dataset with the same values of θ and ϕ.

4 IWEM for missing data

An appealing advantage of IWEM is that it can be modified easily for a nonstandard case whenever the EM algorithm can be so. For illustration, in this section, we propose a modification of IWEM for missing data, called miss-IWEM.

Suppose that a given datum $\mathbf{x}$ is decomposed as $\mathbf{x} = (\mathbf{x}^{(o)}, \mathbf{x}^{(m)})$, where we observe $\mathbf{x}^{(o)}$ but not $\mathbf{x}^{(m)}$. Then $p(\mathbf{x}^{(o)}; \theta) = \int p(\mathbf{x}^{(o)}, \mathbf{x}^{(m)}, \mathbf{z}; \theta) d\mathbf{z} d\mathbf{x}^{(m)}$ is the marginal likelihood of an observed variable $\mathbf{x}^{(o)}$. For simplicity, we assume that $\mathbf{x}$s are independent conditional on $\mathbf{z}$, that means $\mathbf{x}^{(o)}$ and $\mathbf{x}^{(m)}$ are independent given $\mathbf{z}$.

4.1 Choice of Proposal Distribution

When there are missing data, we propose to use the following proposal distribution in P-step of miss-IWEM, which is formulated as

$$q(\mathbf{x}^{(m)}, \mathbf{z} | \mathbf{x}^{(o)}; \theta, \phi) := p(\mathbf{x}^{(m)} | \mathbf{z}; \theta) \cdot q(\mathbf{z} | \breve{\mathbf{x}}; \phi),$$

where $\breve{\mathbf{x}} = (\mathbf{x}^{(o)}, \breve{\mathbf{x}}^{(m)})$ is a completion of the vector $\mathbf{x}$ for some reasonably imputed value $\breve{\mathbf{x}}^{(m)}$ of $\mathbf{x}^{(m)}$ and $q(\mathbf{z}|\mathbf{x}; \phi)$ has the same distribution as q in IWEM. In miss-IWEM, we generate $\breve{\mathbf{x}}^{(m)}$ as follows:

- Draw $\breve{\mathbf{z}}$ from the distribution $q(\mathbf{z}|\tilde{\mathbf{x}}_0; \phi)$,
- and draw $\breve{\mathbf{x}}^{(m)}$ from the distribution $p(\mathbf{x}^{(m)}|\breve{\mathbf{z}}; \theta)$,

where $\tilde{\mathbf{x}}_0 = (\mathbf{x}^{(o)}, \mathbf{0})$ and $\mathbf{0}$ is the 0-vector of dimension equal to that of $\mathbf{x}^{(m)}$.

Algorithm 2: IWEM

Require: Train dataset: $\mathcal{D} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$
Require: Model architectures: $p(\mathbf{x}, \mathbf{z}; \theta)$ and $q(\mathbf{z}|\mathbf{x}; \phi)$
Require: Size of samples: L, K and J
Require: Size of mini-batch: m
Require: SGD based optimization algorithm: $\mathcal{L}(loss, params, current\_params)$
Require: Increment $c > 0$ and number of update criteria n_u
Initialization: Parameters $\theta^{(1)}$ and $\phi^{(1)}$
Initialization: t $\leftarrow 1$
Initialization: Annealing controller $\alpha \leftarrow 0$
while $\theta^{(t)}$ "not converge" **do**
  Sample $\tilde{\mathbf{x}}_1, ..., \tilde{\mathbf{x}}_m$ from $\mathcal{D}$
  E-step Calculate the sum of (8) for mini-batch:

$$\widehat{Q}^{\alpha}(\theta|\theta^{(t)}, \phi^{(t)}) := \sum_{i=1}^{m} \widehat{Q}^{\alpha}(\theta|\theta^{(t)}, \phi^{(t)}; \tilde{\mathbf{x}}_i).$$

  M-step Update $\theta^{(t+1)}$:

$$\theta^{(t+1)} \leftarrow \mathcal{L}\left(-\widehat{Q}^{\alpha}(\theta|\theta^{(t)}, \phi^{(t)}), \theta, \theta^{(t)}\right).$$

  P-step Calculate the sum of (7) for mini-batch and update $\phi^{(t+1)}$:

$$\phi^{(t+1)} \leftarrow \mathcal{L}\left(-\sum_{i=1}^{m} \widehat{L}^{\text{opt}}(\theta^{(t+1)}, \phi; \tilde{\mathbf{x}}_i), \phi, \phi^{(t)}\right).$$

  After every n_u updates, $\alpha \leftarrow \min(\alpha + c, 1)$
end

4.2 miss-IWEM

In E-step, likewise IWEM, we consider *annealing strategy* with an annealing controller α and calculate the following objective function:

$$\begin{aligned}\widehat{Q}^{\alpha}_{\text{m}}(\theta|\theta^{(t)}, \phi^{(t)}; \mathbf{x}^{(o)}) \quad := \quad & \alpha \cdot \widehat{Q}_{\text{m}}(\theta|\theta^{(t)}, \phi^{(t)}; \mathbf{x}^{(o)}) \\ & + (1-\alpha) \cdot \widehat{L}^{\text{VAE}}_{\text{m}}(\theta, \phi^{(t)}; \mathbf{x}^{(o)}).\end{aligned} \tag{9}$$

Here we define

$$\widehat{Q}_{\text{m}}(\theta|\theta^{(t)}, \phi^{(t)}; \mathbf{x}^{(o)}) = \sum_{k=1}^{K} \frac{w_k}{\sum_{k'} w_{k'}} \log p(\mathbf{x}^{(o)}, \mathbf{x}_k^{(m)}, \mathbf{z}_k; \theta),$$

where $(\mathbf{x}_k^{(m)}, \mathbf{z}_k) \sim q(\mathbf{x}^{(m)}, \mathbf{z}|\mathbf{x}^{(o)}; \theta^{(t)}, \phi^{(t)})$,

$$w_k := \frac{p(\mathbf{x}^{(o)}, \mathbf{x}_k^{(m)}, \mathbf{z}_k; \theta^{(t)})}{q(\mathbf{x}_k^{(m)}, \mathbf{z}_k|\mathbf{x}^{(o)}; \theta^{(t)}, \phi^{(t)})} = \frac{p(\mathbf{x}^{(o)}, \mathbf{z}_k; \theta^{(t)})}{q(\mathbf{z}_k|\breve{\mathbf{x}}; \phi^{(t)})}$$

for $k = 1, ..., K$, and

$$\widehat{L}_{\mathrm{m}}^{\mathrm{VAE}}(\theta, \phi^{(t)}; \mathbf{x}^{(o)}) := \frac{1}{L} \sum_{l=1}^{L} \log \left(\frac{p(\mathbf{x}^{(o)}, \mathbf{z}_l; \theta)}{q(\mathbf{z}_l | \breve{\mathbf{x}};, \phi^{(t)})} \right),$$

where $\mathbf{z}_l \sim q(\mathbf{z}|\breve{\mathbf{x}}; \phi^{(t)})$ for $l = 1, ..., L$.

After updating θ to $\theta^{(t+1)}$ with the objective function (9), we also apply *optimal P-step* technique. The objective function of P-step is written as

$$\begin{aligned} &\widehat{L}_{\mathrm{m}}^{\mathrm{opt}}(\theta^{(t+1)}, \phi; \mathbf{x}^{(o)}) \\ &:= \log \left(\frac{1}{J} \sum_{j=1}^{J} \frac{\tilde{p}(\mathbf{x}_j^{(m)}, \mathbf{z}_j | \mathbf{x}^{(o)}; \theta^{(t+1)})}{q(\mathbf{x}_j^{(m)}, \mathbf{z}_j | \mathbf{x}^{(o)}; \theta^{(t+1)}, \phi)} \right), \end{aligned} \tag{10}$$

where $(\mathbf{x}_j^{(m)}, \mathbf{z}_j) \sim q(\mathbf{x}^{(m)}, \mathbf{z}|\mathbf{x}^{(o)}; \theta^{(t+1)}, \phi)$ for $j = 1, ..., J$ and

$$\begin{aligned} \tilde{p}(\mathbf{x}^{(m)}, \mathbf{z}|\mathbf{x}^{(o)}; \theta^{(t+1)}) \quad \propto \quad & |\log p(\mathbf{x}^{(o)}, \mathbf{x}^{(m)}, \mathbf{z}; \theta^{(t+1)})| \\ & \times p(\mathbf{x}^{(o)}, \mathbf{x}^{(m)}, \mathbf{z}; \theta^{(t+1)}). \end{aligned}$$

By canceling out the term $p(\mathbf{x}_j^{(m)}|\mathbf{z}_j; \theta^{(t+1)})$, (10) can be rewritten as

$$\log \left(\frac{1}{J} \sum_{j=1}^{J} \frac{|\log p(\mathbf{x}^{(o)}, \mathbf{x}_j^{(m)}, \mathbf{z}_j; \theta^{(t+1)})| \cdot p(\mathbf{x}^{(o)}, \mathbf{z}_j; \theta^{(t+1)})}{q(\mathbf{z}_j|\breve{\mathbf{x}}; \phi)} \right).$$

The pseudo algorithm of miss-IWEM is summarized in Algorithm 3.

Algorithm 3: miss-IWEM

Require: Train dataset: $\mathcal{D} = \{\mathbf{x}_1^{(o)}, ..., \mathbf{x}_n^{(o)}\}$
Require: Other requirements are same as Algorithm 2
Initialization: Same as Algorithm 2
while $\theta^{(t)}$ "not converge" **do**
  Sample $\tilde{\mathbf{x}}_1^{(o)}, ..., \tilde{\mathbf{x}}_m^{(o)}$ from $\mathcal{D}$
  E-step Calculate the sum of (9) for mini-batch:

$$\widehat{Q}_{\mathrm{m}}^{\alpha}(\theta|\theta^{(t)}, \phi^{(t)}) := \sum_{i=1}^{m} \widehat{Q}_{\mathrm{m}}^{\alpha}(\theta|\theta^{(t)}, \phi^{(t)}; \tilde{\mathbf{x}}_i^{(o)}).$$

  M-step Update $\theta^{(t+1)}$:

$$\theta^{(t+1)} \leftarrow \mathcal{L}\left(-\widehat{Q}_{\mathrm{m}}^{\alpha}(\theta|\theta^{(t)}, \phi^{(t)}), \theta, \theta^{(t)}\right).$$

  P-step Calculate the sum of (10) for mini-batch and update $\phi^{(t+1)}$:

$$\phi^{(t+1)} \leftarrow \mathcal{L}\left(-\sum_{i=1}^{m} \widehat{L}_{\mathrm{m}}^{\mathrm{opt}}(\theta^{(t+1)}, \phi; \tilde{\mathbf{x}}_i^{(o)}), \phi, \phi^{(t)}\right).$$

  After every n_u updates, $\alpha \leftarrow \min(\alpha + c, 1)$
end

5 Empirical analysis

5.1 Experimental setup

Datasets We carry out various experimental analyses to assess the performances of IWEM and miss-IWEM in comparison to other learning methods by analyzing four image datasets: static biMNIST [Larochelle and Murray, 2011], dynamic biMNIST [Salakhutdinov and Murray, 2008], Omniglot [Lake et al., 2015] and Caltech 101 Silhouette [Marlin et al., 2010]. For missing data experiments, we additionally analyze four UCI datasets: Bank, Breast, Red and White.

Model architectures For image datasets, we consider MLP and CNN architectures for $(p(\mathbf{x}|\mathbf{z}; \theta), q(\mathbf{z}|\mathbf{x}; \phi))$ and use the Gaussian distribution for the proposal family. We exploit the same settings implemented in Tucker et al. [2018] for MLPs and refer to Tomczak and Welling [2017] for the details of CNNs. We use the Gaussian distribution for the proposal distribution family. For UCI datasets, we consider MLPs with 3 hidden layers (with 128 hidden units) and tanh activations which are used in Mattei and Frellsen [2018]. We also use the Gaussian distribution for the proposal distribution family.

Implementation details IWEM has the three tuning parameters, L, K and J. We find in practice that L does not affect the performances seriously and thus we fix L to be 1 for computational efficiency. We choose the optimal K and J using validation datasets. For the annealing scheme, we start with the annealing controller α being zero and increase it by 0.01 after every epoch of the training phase up to 1. For the optimization algorithm, the *Adam* algorithm [Kingma and Ba, 2014] is used with the learning rate $5 \cdot 10^{-4}$ and mini-batches of size 100. The initial values of the parameters of the generative and proposal models are designed according to [Glorot and Bengio, 2010].

5.2 Complete data analysis

Results We consider three algorithms: 1) IWEM without *optimal P-step* (IWEM-woo), 2) IWEM without *annealing strategy* (IWEM-woa) and 3) IWEM. We conduct the test log-likelihood comparisons which are calculated by the same way used in Burda et al. [2015], Rezende et al. [2014], Tomczak and Welling [2017]. We train each model 5 times, take average of the test log-likelihood values and compare ours with other methods including STL [Roeder et al., 2017] and DReG [Tucker et al., 2018], which are summarized in Table 1.

Using *optimal P-step* or *annealing strategy* improves

Table 1: Test log-likelihood for diverse learning methods. (STL and DReG are implemented with the public github code of DReG paper.)

MLP	s.MNIST	d.MNIST	Omni.	Caltech.
IWAE	-88.86	-87.64	-107.23	-124.31
STL	-	-87.43	-106.40	-
IWAE-DReG	-	-87.76	-106.70	-
IWEM-woo	-87.97	-86.99	-106.59	-123.66
IWEM-woa	**-87.59**	**-86.61**	-106.5	-124.19
IWEM	-87.71	-86.68	**-106.20**	**-123.50**
CNN	s.MNIST	d.MNIST	Omni.	Caltech.
VAE	-84.63	-84.08	-101.63	-109.24
IWAE	-83.54	-81.56	-100.27	-106.94
IWEM-woo	-83.84	-81.32	-100.24	-106.65
IWEM-woa	**-83.32**	**-81.07**	**-100.15**	-106.19
IWEM	-83.77	-81.28	-100.39	**-106.05**

IWAE and either IWEM or IWEM-woa always achieves the highest test log-likelihood values for all considering cases. These results suggest that *optimal P-step* and *annealing strategy* are helpful to reduce the variance of the objective function in E-step, leading to better performance.

We observe that *annealing strategy* is not always compatible with *optimal P-step*. One of possible explanations would be due to the bias raised by the ELBO function. The objective function $\widehat{Q}^{\alpha}$ in (8) has small variance but has large bias at early learning stages. And this large bias may prevent the learning algorithm from searching a good solution. However, in the next section, we will demonstrate that *annealing strategy* is essential for missing data cases, which is partly because the variance of $\widehat{Q}^{\alpha}_{\mathrm{m}}$ is quite large even at early learning stages.

Ablation study We conduct an ablation study to investigate the sensitivity of performances with respect to the choices of K and J. The results are reported in Table 2. K does not affect to the performances much unless it is too small. In contrast, the performances are sensitive to the choice of J. The log-likelihood values keep increasing until J reaches 50. This observation indicates that the new divergence (7) works well for a relatively large J.

We verify whether the deep generative model trained by IWEM generates realistic images whose results are depicted in the supplementary material. The figure illustrates that IWEM is also good at image generation.

5.3 Incomplete data analysis

Incomplete data generation strategy For image analysis, Mattei and Frellsen [2018] considered the scenario where one observes pixels of images uniformly at random. But it is difficult to assess relative performances of competing methods because images with missing pixels generated by this manner are still easily recognizable. In this study we consider a more difficult scenario, that is, we only observe patch-wise-removed images. We generate incomplete images as follows (Figure 3 visualizes this procedure more clearly.):

- Divide an image into nine equal patches,
- and generate an incomplete image by removing the predefined number of patches randomly.

For UCI datasets, we corrupt each data by removing half of the features uniformly at random as is done by Mattei and Frellsen [2018].

Table 2: Test log-likelihood of static biMNIST for various values of K and J. The other parameters on each case is fixed to the optimal value.

K	1	5	10	20	50	70
LL	-89.38	-87.30	-87.13	-87.14	**-87.11**	-87.16
J	1	5	10	20	50	70
LL	-90.31	-88.16	-87.76	-87.34	**-87.11**	-87.20

Results We consider two algorithms: 1) miss-IWEM and 2) miss-IWEM without *annealing strategy* (miss-IWEM-woa). We compute the test log-likelihood values over the static biMNIST dataset and the mean-squared errors for imputation of features for UCI datasets. We compare our proposed methods with missIWAE [Mattei and Frellsen, 2018]. For all cases we estimate each model 5 times, take average the resulted values and compare our methods with missIWAE [Mattei and Frellsen, 2018], which are reported in Table 3 and 4.

Our proposed methods consistently outperform missIWAE for all cases. Moreover as can be seen in Table 3, the margins between ours and missIWAE become larger as the number of cropped patches increases.

We also want to stress that *annealing strategy* for miss-IWEM is always helpful, which contrasts sharply to the complete cases in Section 5.2 where *annealing strategy* is not must. This result would be because of larger variance due to missing data. In missing data case we need to sample the missing vector $\mathbf{x}^{(m)}$ as well as the latent vector $\mathbf{z}$ to approximate the objective function in the E-step, and this additional procedure increases the variance of snIS. And the variance is expected to become larger as the missing rate increases, which is empirically confirmed in Figure 4. Thus utilizing the ELBO function to reduce the variance is necessary

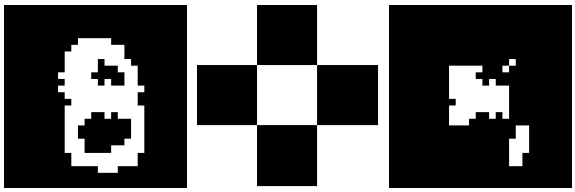

Figure 3: Example of (**left**) an original image, (**middle**) 4 randomly cropped patches coloured black and (**right**) the resulted incomplete image.

Table 3: Test log-likelihood values for various missing scenarios on the static biMNIST.

# of cropped patches	missIWAE	miss-IWEM-woa	miss-IWEM
3	-90.29	-89.79	**-89.71**
4	-92.07	-90.97	**-90.76**
5	-95.54	-93.33	**-92.23**
6	-102.26	-97.66	**-95.18**

to stabilize the learning procedure at early iterations despite of adding some biases.

Ablation study We also visualize the power of miss-IWEM for completion of missing data. After completing the learning procedure of θ and ϕ with static biMNIST dataset, we impute missing patches by use of the imputation technique described in Section 4.1. Figure 5 shows that miss-IWEM also works well in image completion task.

6 Conclusion

In this study we proposed a new and general approach to learn deep generative models, called IWEM, which improves IWAE. Based on the interpretation that IWAE can be understood as an EM algorithm, we devised two new techniques to reduce the variance due to snIS in E-step. In addition we modified IWEM for missing data, called miss-IWEM. We demonstrated empirically that our methods are superior in terms of the test log-likelihood compared to the recent methods as well as IWAE, for both fully-observed data and missing data scenarios. Especially we observed that the margin of miss-IWEM compared to missIWAE becomes larger as the missing rate increases.

Miss-IWEM can be applied to the case where the number of cropped patches are different in each images. One important application is to train a deep generative model based on image data with different resolutions whose results will be reported elsewhere soon.

It is an interesting future work to apply IWEM to the disentanglement problem [Achille and Soatto, 2018, Chen et al., 2018, Higgins et al., 2017, Kim and Mnih, 2018]. We expect that replacing the reconstruction loss in the variational based methods by the E-step loss function in IWEM would result in a better trade-off between density estimation and disentanglement.

Table 4: Mean-squared errors for imputation for various UCI datasets.

	Bank	Breast	Red	White
missIWAE	0.598	0.351	0.537	0.566
miss-IWEM	**0.465**	**0.346**	**0.483**	**0.479**

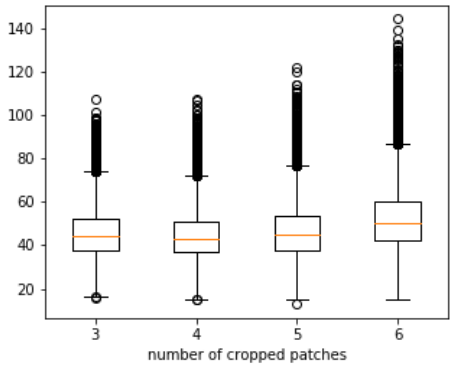

Figure 4: Boxplots of variances of $\widehat{Q}_{\mathrm{m}}^{\alpha}$ in (9) with $\alpha = 1$ for various numbers of cropped patches.

Figure 5: Completion of 3 incomplete images, which are obtained by cropping 6 patches at random, by miss-IWEM. (**1st column**) Observed incomplete images, (**2nd column**) ground-truth images and (**3rd column**) imputed images.

Acknowledgements

This work is supported by Samsung Electronics Co., Ltd.

References

Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. J. Mach. Learn. Res., 19(1):1947–1980, January 2018. ISSN 1532-4435.

Thomas Bengtsson, Peter Bickel, and Bo Li. Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. 2008.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. arXiv preprint arXiv:1511.06349, 2015.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. arXiv preprint arXiv:1509.00519, 2015.

Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems 31, pages 2610–2620. Curran Associates, Inc., 2018.

Chris Cremer, Quaid Morris, and David Duvenaud. Reinterpreting importance-weighted autoencoders. arXiv preprint arXiv:1704.02916, 2017.

Adji B Dieng and John Paisley. Reweighted expectation maximization. arXiv preprint arXiv:1906.05850, 2019.

Matthew Dowling, Josue Nassar, Petar M Djurić, and Mónica F Bugallo. Improved adaptive importance sampling based on variational inference. In 2018 26th European Signal Processing Conference (EUSIPCO), pages 1632–1636. IEEE, 2018.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics, 2010.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In International Conference on Learning Representations, volume 3, 2017.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In Jennifer Dy and Andreas Krause, editors, Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pages 2649–2658, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

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

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

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, Advances in Neural Information Processing Systems 29, pages 4743–4751. Curran Associates, Inc., 2016.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. Science, 350(6266): 1332–1338, 2015.

Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pages 29–37, 2011.

Benjamin Marlin, Kevin Swersky, Bo Chen, and Nando Freitas. Inductive principles for restricted boltzmann machine learning. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pages 509–516, 2010.

Pierre-Alexandre Mattei and Jes Frellsen. missiwae: Deep generative modelling and imputation of incomplete data. arXiv preprint arXiv:1812.02633, 2018.

Art B. Owen. Monte Carlo theory, methods and examples. 2013.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Francis Bach and David Blei, editors, Proceedings of the 32nd International Conference on Machine Learning, volume 37 of Proceedings of Machine Learning Research, pages 1530–1538, Lille, France, 07–09 Jul 2015. PMLR.

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

Geoffrey Roeder, Yuhuai Wu, and David K Duvenaud. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In Advances in Neural Information Processing Systems, pages 6925–6934, 2017.

Ruslan Salakhutdinov and Iain Murray. On the quantitative analysis of deep belief networks. In Proceedings of the 25th International Conference on Machine Learning, ICML '08, pages 872–879, 2008. ISBN 978-1-60558-205-4.

Tim Salimans, Diederik Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap. In International Conference on Machine Learning, pages 1218–1226, 2015.

Casper Kaae Sø nderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sø nderby, and Ole Winther. Ladder variational autoencoders. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, Advances in Neural Information Processing Systems 29, pages 3738–3746. Curran Associates, Inc., 2016.

Zhao Song, Ricardo Henao, David Carlson, and Lawrence Carin. Learning sigmoid belief networks via monte carlo expectation maximization. In Artificial Intelligence and Statistics, pages 1347–1355, 2016.

Surya T Tokdar and Robert E Kass. Importance sampling: a review. Wiley Interdisciplinary Reviews: Computational Statistics, 2(1):54–60, 2010.

Jakub M Tomczak and Max Welling. Vae with a vampprior. arXiv preprint arXiv:1705.07120, 2017.

George Tucker, Dieterich Lawson, Shixiang Gu, and Chris J Maddison. Doubly reparameterized gradient estimators for monte carlo objectives. arXiv preprint arXiv:1810.04152, 2018.