Deep latent variable models: estimation and missing data imputation

Jes Frellsen

Department of Computer Science
IT University of Copenhagen

https://frellsen.org
@jesfrellsen

February 8, 2019
DIKU

Joint work with
Pierre-Alexandre Mattei
(ITU)

Overview of talk

Introduction to Deep Latent Variable Models (DLVM)

Contribution 1: On the boundedness of the likelihood of DLVMs

Contribution 2: Missing data imputation using the exact conditional distribution

Contribution 3: Training and imputation on incomplete data sets

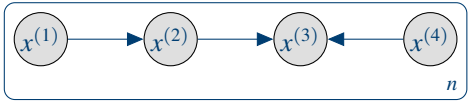
Probabilistic generative and latent variable models

Mohamed and Rezende (2017)

A **generative model**:

> "Describes a process that is assumed to rise to some data" (MacKay, 2003).

- In **unsupervised** learning, we model the (joint) **density** $p(\mathbf{x})$.
- The model allows for **generating data**: i.e. $\mathbf{x} \sim p(\mathbf{x})$
- Typically we assume some **factorisation** of $p(\mathbf{x})$, e.g.



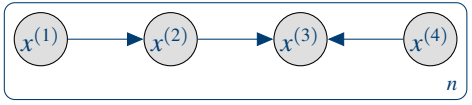
Probabilistic generative and latent variable models

Mohamed and Rezende (2017)

A **generative model**:

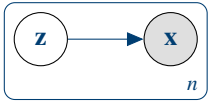
> "Describes a process that is assumed to rise to some data" (MacKay, 2003).

- In **unsupervised** learning, we model the (joint) **density** $p(\mathbf{x})$.
- The model allows for **generating data**: i.e. $\mathbf{x} \sim p(\mathbf{x})$
- Typically we assume some **factorisation** of $p(\mathbf{x})$, e.g.



In a **latent variable model** we assume that there is an **unobserved random variable** $\mathbf{z}$.

- The **joint density** $p(\mathbf{x}, \mathbf{z})$ is modelled
- The model allows for **generating** $\mathbf{x}, \mathbf{z} \sim p(\mathbf{x}, \mathbf{z})$.
- Typically we assume some **factorization** of $p(\mathbf{x}, \mathbf{z})$, e.g.



- **Dimension reduction:** we can think of $\mathbf{z}$ as a **code** summarizing multivariate data $\mathbf{x}$.

Deep latent variable models (DLVMs)

Mohamed and Rezende (2017)

Probabilistic generative models:

- $+$ Well founded framework for model building, inference and prediction.
- $\div$ Limited complexity: Mainly conjugate and linear models
- $\div$ Learning is computational expensive

Deep latent variable models (DLVMs)

Mohamed and Rezende (2017)

Probabilistic generative models:

- \+ Well founded framework for model building, inference and prediction.
- ÷ Limited complexity: Mainly conjugate and linear models
- ÷ Learning is computational expensive

Deep neural networks:

- \+ Rich non-linear models
- \+ Scalable learning using stochastic gradient decent
- ÷ Only give point estimates
- ÷ Typically discriminative and non-probabilistic

Deep latent variable models (DLVMs)

Mohamed and Rezende (2017)

Probabilistic generative models:

- \+ Well founded framework for model building, inference and prediction.
- ÷ Limited complexity: Mainly conjugate and linear models
- ÷ Learning is computational expensive

Deep neural networks:

- \+ Rich non-linear models
- \+ Scalable learning using stochastic gradient decent
- ÷ Only give point estimates
- ÷ Typically discriminative and non-probabilistic

Deep latent variable models combine the approximation abilities of deep neural networks and the statistical foundations of generative models.

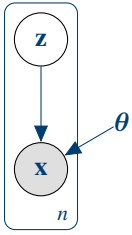
Independently invented by Kingma and Welling (2014), as **variational autoencoders**, and Rezende et al. (2014) as **deep latent Gaussian models**.

Deep latent variable models (DLVMs)

Kingma and Welling (2014) and Rezende et al. (2014)

Assume that $(\mathbf{x}_i, \mathbf{z}_i)_{i \leq n}$ are i.i.d. random variables driven by the model:

$$\begin{cases} \mathbf{z} \sim p(\mathbf{z}) & \text{(prior)} \\ \mathbf{x} \sim p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) & \text{(observation model)} \end{cases}$$



where

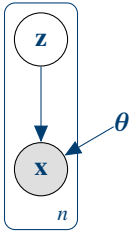
- $\mathbf{z} \in \mathbb{R}^d$ is the **latent** variable,
- $\mathbf{x} \in \mathcal{X}$ is the **observed** variable.

Deep latent variable models (DLVMs)

Kingma and Welling (2014) and Rezende et al. (2014)

Assume that $(\mathbf{x}_i, \mathbf{z}_i)_{i \leq n}$ are i.i.d. random variables driven by the model:

$$\begin{cases} \mathbf{z} \sim p(\mathbf{z}) & \text{(prior)} \\ \mathbf{x} \sim p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) = \Phi(\mathbf{x} \mid f_{\boldsymbol{\theta}}(\mathbf{z})) & \text{(observation model)} \end{cases}$$



where

- $\mathbf{z} \in \mathbb{R}^d$ is the **latent** variable,
- $\mathbf{x} \in \mathcal{X}$ is the **observed** variable,
- the function $f_{\boldsymbol{\theta}} : \mathbb{R}^d \to H$ is a **(deep) neural network** called the **decoder**,
- $(\Phi(\cdot \mid \boldsymbol{\eta}))_{\boldsymbol{\eta} \in H}$ is a parametric family called the **output density**.

  The output density is usually **very simple**: unimodal and fully factorised (e.g. multivariate Gaussians or products of multinomials).

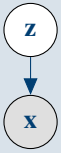
The role of the prior

As in regular factor analysis, the prior distribution of the latent variable is often an **isotropic Gaussian** $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}_d, \mathbf{I}_d)$.

Factor analysis (FA)

In **FA** the generative process is:

- $\mathbf{z} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$,
- $\mathbf{x}|\mathbf{z} \sim \mathcal{N}(\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \boldsymbol{\Psi})$.



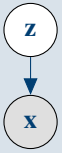
The role of the prior

As in regular factor analysis, the prior distribution of the latent variable is often an **isotropic Gaussian** $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}_d, \mathbf{I}_d)$.

Factor analysis (FA)

In **FA** the generative process is:

- $\mathbf{z} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$,
- $\mathbf{x}|\mathbf{z} \sim \mathcal{N}(\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \boldsymbol{\Psi})$.



More complex, **learnable priors** have also been considered.

For example, in Harchaoui et al. (2018) looked at **mixtures of K Gaussians**:

$$p(\mathbf{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

where the parameters $\pi_1, ..., \pi_K, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, ...\boldsymbol{\Sigma}_K$ are learned.

The role of the decoder

The role of the **decoder** $f_\theta : \mathbb{R}^d \to H$ is:

- To transform $\mathbf{z}$ (**the code**) into parameters $\eta = f_\theta(\mathbf{z})$ of the output density $\Phi(\cdot \mid \eta)$.
- The weights θ of the **decoder** are learned.

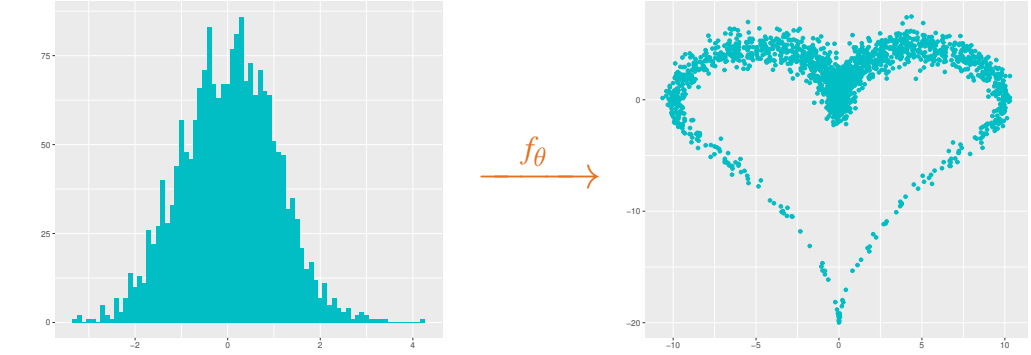
The role of the decoder

The role of the **decoder** $f_\theta : \mathbb{R}^d \to H$ is:

- To transform $\mathbf{z}$ (**the code**) into parameters $\eta = f_\theta(\mathbf{z})$ of the output density $\Phi(\cdot \mid \eta)$.
- The weights θ of the **decoder** are learned.

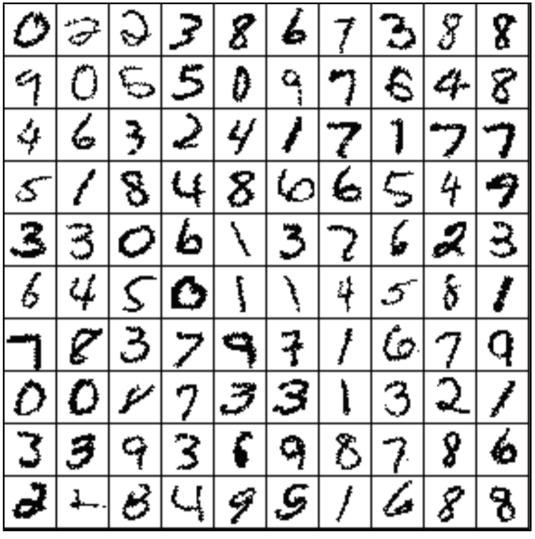
Illustrative example of a simple non-linear decoder ($d = 1$, $p = 2$, Gaussian outputs).

Here, f_θ **transforms** $\mathbf{z}$ **into both a mean and a covariance matrix.**

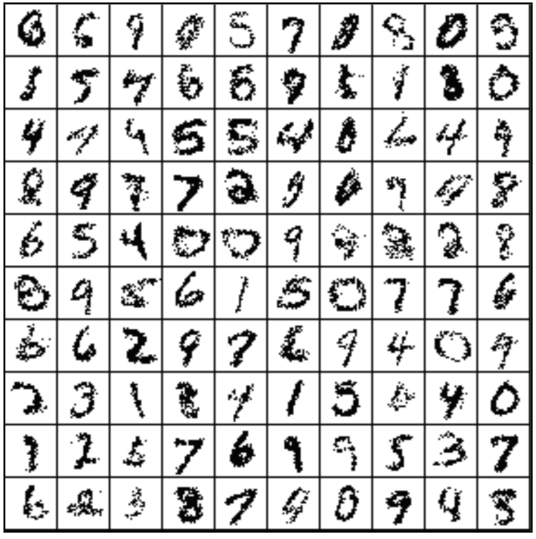


DLVMs applications: density estimation on MNIST

Rezende et al. (2014)



Training data



Model samples

DLVMs applications: density estimation on (Brendan) Frey faces

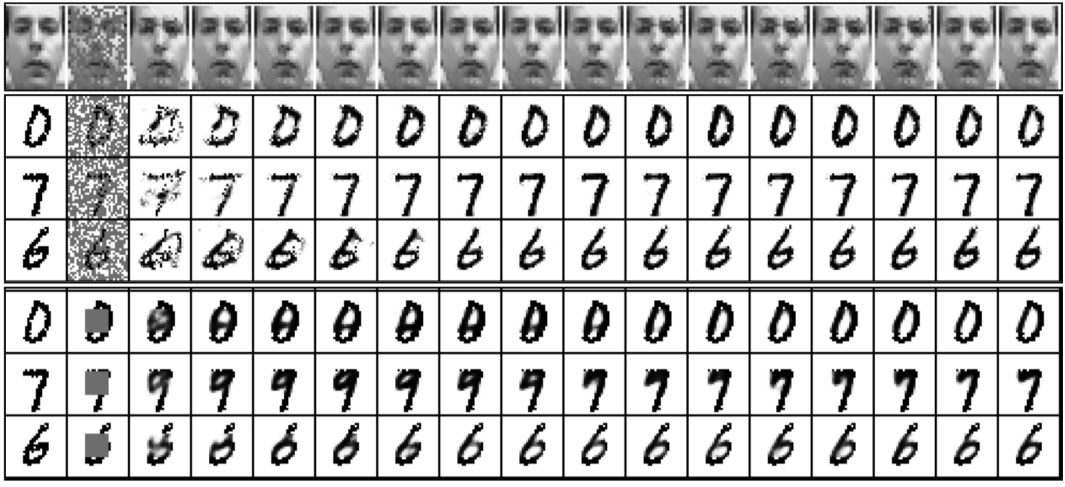
Rezende et al. (2014)

Training data

Model samples

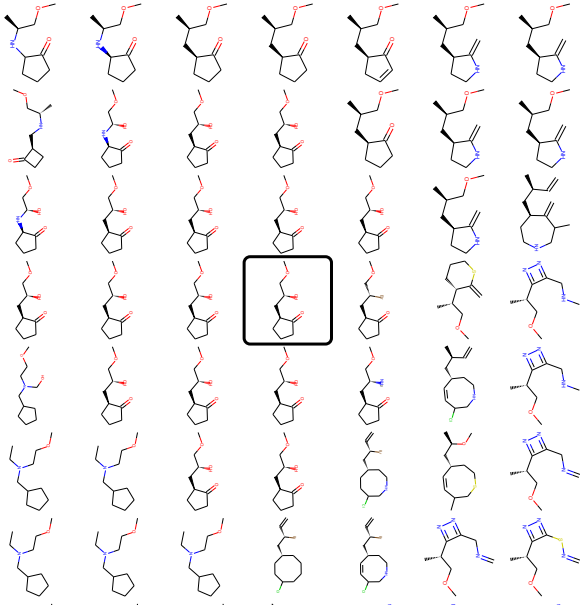
DLVMs applications: missing data imputation

Rezende et al. (2014)



DLVMs applications: molecular design

Grammar Variational Autoencoder by Kusner et al. (2017)



Learning DLVMs

Kingma and Welling (2014) and Rezende et al. (2014)

Given a data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\intercal \in \mathcal{X}^n$, the **log-likelihood function** for a DLVM is

$$\ell(\boldsymbol{\theta}) = \log \underbrace{p_{\boldsymbol{\theta}}(\mathbf{X})}_{p(\mathbf{X}|\boldsymbol{\theta})} = \sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}(\mathbf{x}_i) \qquad \text{where} \qquad p_{\boldsymbol{\theta}}(\mathbf{x}_i) = \int_{\mathbb{R}^d} p_{\boldsymbol{\theta}}(\mathbf{x}_i \mid \mathbf{z}) p(\mathbf{z}) \, \mathrm{d}\mathbf{z}.$$

We would like to find a **MLE,** $\hat{\boldsymbol{\theta}} \in \arg\max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta})$.

Learning DLVMs

Kingma and Welling (2014) and Rezende et al. (2014)

Given a data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\intercal \in \mathcal{X}^n$, the **log-likelihood function** for a DLVM is

$$\ell(\boldsymbol{\theta}) = \log \underbrace{p_{\boldsymbol{\theta}}(\mathbf{X})}_{p(\mathbf{X}|\boldsymbol{\theta})} = \sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}(\mathbf{x}_i) \qquad \text{where} \qquad p_{\boldsymbol{\theta}}(\mathbf{x}_i) = \int_{\mathbb{R}^d} p_{\boldsymbol{\theta}}(\mathbf{x}_i \mid \mathbf{z}) p(\mathbf{z}) \, \mathrm{d}\mathbf{z}.$$

We would like to find a **MLE,** $\hat{\boldsymbol{\theta}} \in \arg\max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta})$.

However, for complicated output density $p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})$

- $p_{\boldsymbol{\theta}}(\mathbf{x})$ is **intractable** rendering direct **MLE intractable**

Learning DLVMs

Kingma and Welling (2014) and Rezende et al. (2014)

Given a data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\intercal \in \mathcal{X}^n$, the **log-likelihood function** for a DLVM is

$$\ell(\boldsymbol{\theta}) = \log \underbrace{p_{\boldsymbol{\theta}}(\mathbf{X})}_{p(\mathbf{X}|\boldsymbol{\theta})} = \sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}(\mathbf{x}_i) \qquad \text{where} \qquad p_{\boldsymbol{\theta}}(\mathbf{x}_i) = \int_{\mathbb{R}^d} p_{\boldsymbol{\theta}}(\mathbf{x}_i \mid \mathbf{z}) p(\mathbf{z}) \, \mathrm{d}\mathbf{z}.$$

We would like to find a **MLE,** $\hat{\boldsymbol{\theta}} \in \arg\max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta})$.

However, for complicated output density $p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})$

- $p_{\boldsymbol{\theta}}(\mathbf{x})$ is **intractable** rendering direct **MLE intractable**
- $p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x})$ is **intractable** rendering **EM intractable**

Learning DLVMs

Kingma and Welling (2014) and Rezende et al. (2014)

Given a data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\intercal \in \mathcal{X}^n$, the **log-likelihood function** for a DLVM is

$$\ell(\boldsymbol{\theta}) = \log \underbrace{p_{\boldsymbol{\theta}}(\mathbf{X})}_{p(\mathbf{X}|\boldsymbol{\theta})} = \sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}(\mathbf{x}_i) \qquad \text{where} \qquad p_{\boldsymbol{\theta}}(\mathbf{x}_i) = \int_{\mathbb{R}^d} p_{\boldsymbol{\theta}}(\mathbf{x}_i \mid \mathbf{z}) p(\mathbf{z}) \, \mathrm{d}\mathbf{z}.$$

We would like to find a **MLE,** $\hat{\boldsymbol{\theta}} \in \arg\max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta})$.

However, for complicated output density $p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})$

- $p_{\boldsymbol{\theta}}(\mathbf{x})$ is **intractable** rendering direct **MLE intractable**
- $p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x})$ is **intractable** rendering **EM intractable**
- **Stochastic EM is not scalable** to large n and moderate d

Variational Inference (VI)

Kingma and Welling (2014), Rezende et al. (2014), and Blei et al. (2017)

VI approximatively maximises the log-likelihood by maximising the **evidence lower bound**

$$\mathcal{L}(\boldsymbol{\theta}, Q) = \mathbb{E}_{\mathbf{z}\sim Q}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{Z})}{Q(\mathbf{Z})}\right] = \ell(\boldsymbol{\theta}) - \mathrm{KL}(Q \,||\, p_{\boldsymbol{\theta}}(\mathbf{Z} \mid \mathbf{X})) \leq \ell(\boldsymbol{\theta})$$

wrt. $(\boldsymbol{\theta}, Q)$, where

- Q **is the variational distribution** over the space of codes $\mathbb{R}^{n\times d}$,
- $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\intercal \in \mathcal{X}^n$ and $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)^\intercal \in \mathbb{R}^{n\times d}$.

Variational Inference (VI)

Kingma and Welling (2014), Rezende et al. (2014), and Blei et al. (2017)

VI approximatively maximises the log-likelihood by maximising the **evidence lower bound**

$$\mathcal{L}(\boldsymbol{\theta}, Q) = \mathbb{E}_{\mathbf{z}\sim Q}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{Z})}{Q(\mathbf{Z})}\right] = \ell(\boldsymbol{\theta}) - \mathrm{KL}(Q \,||\, p_{\boldsymbol{\theta}}(\mathbf{Z} \mid \mathbf{X})) \leq \ell(\boldsymbol{\theta})$$

wrt. $(\boldsymbol{\theta}, Q)$, where

- Q **is the variational distribution** over the space of codes $\mathbb{R}^{n\times d}$,
- $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\intercal \in \mathcal{X}^n$ and $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)^\intercal \in \mathbb{R}^{n\times d}$.

Normally in **VI**, Q would come for a parametric family of distribution $(Q_{\mathbf{m}})_{\mathbf{m}\in\mathcal{M}}$, and we would maximise $\mathcal{L}(\boldsymbol{\theta}, Q_{\mathbf{m}})$ wrt. $(\boldsymbol{\theta}, \mathbf{m})$.

However, computing a distribution over $\mathbb{R}^{n\times d}$ is too expensive for large datasets.

Amortised Variational Inference (AVI)

Kingma and Welling (2014) and Rezende et al. (2014)

Amortised inference scales up VI by

(a) Writing Q as a product of conditional variational distributions:

$$Q_{\gamma}(\mathbf{Z}) = \prod_{i=1}^{n} q_{\gamma}(\mathbf{z}_i|\mathbf{x}_i)$$

(b) Learning a function g_{γ} that **transforms each data point $\mathbf{x}_i$ into the parameters of the corresponding conditional** $q(\mathbf{z}_i|\mathbf{x}_i) = \Psi(\mathbf{z}_i|g_{\gamma}(\mathbf{x}_i))$.

[1] Stochastic gradients of $\mathcal{L}(\boldsymbol{\theta}, Q_{\gamma})$ can be calculated using the reparameterization trick.

Amortised Variational Inference (AVI)

Kingma and Welling (2014) and Rezende et al. (2014)

Amortised inference scales up VI by

(a) Writing Q as a product of conditional variational distributions:

$$Q_\gamma(\mathbf{Z}) = \prod_{i=1}^{n} q_\gamma(\mathbf{z}_i|\mathbf{x}_i)$$

(b) Learning a function g_γ that **transforms each data point $\mathbf{x}_i$ into the parameters of the corresponding conditional** $q(\mathbf{z}_i|\mathbf{x}_i) = \Psi(\mathbf{z}_i|g_\gamma(\mathbf{x}_i))$.

The family $(\Psi(\cdot|\kappa))_{\kappa\in\mathcal{K}}$ is a parametric family of distributions over $\mathbb{R}^d$ (usually Gaussians), $g_\gamma : \mathcal{X} \to K$ is a neural net called the **inference network.**

[1] Stochastic gradients of $\mathcal{L}(\boldsymbol{\theta}, Q_\gamma)$ can be calculated using the reparameterization trick.

Amortised Variational Inference (AVI)

Kingma and Welling (2014) and Rezende et al. (2014)

Amortised inference scales up VI by

(a) Writing Q as a product of conditional variational distributions:

$$Q_{\gamma}(\mathbf{Z}) = \prod_{i=1}^{n} q_{\gamma}(\mathbf{z}_i|\mathbf{x}_i)$$

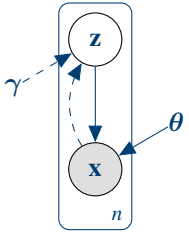
(b) Learning a function g_{γ} that **transforms each data point $\mathbf{x}_i$ into the parameters of the corresponding conditional** $q(\mathbf{z}_i|\mathbf{x}_i) = \Psi(\mathbf{z}_i|g_{\gamma}(\mathbf{x}_i))$.

The family $(\Psi(\cdot|\kappa))_{\kappa\in\mathcal{K}}$ is a parametric family of distributions over $\mathbb{R}^d$ (usually Gaussians), $g_{\gamma} : \mathcal{X} \to K$ is a neural net called the **inference network.**

Inference for DLVMs solves the optimisation problem

$$\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta},\boldsymbol{\gamma}\in\boldsymbol{\Gamma}} \mathcal{L}(\boldsymbol{\theta}, Q_{\gamma}),$$

typically using **stochastic gradient descent**.[1]



A DLVM combined with AVI is called a **variational autoencoder (VAE).**

[1] Stochastic gradients of $\mathcal{L}(\boldsymbol{\theta}, Q_{\gamma})$ can be calculated using the reparameterization trick.

Amortised Variational Inference (AVI)

Kingma and Welling (2014) and Rezende et al. (2014)

Amortised inference scales up VI by

(a) Writing Q as a product of conditional variational distributions:

$$Q_\gamma(\mathbf{Z}) = \prod_{i=1}^{n} q_\gamma(\mathbf{z}_i|\mathbf{x}_i)$$

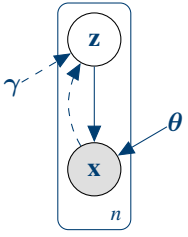
(b) Learning a function g_γ that **transforms each data point $\mathbf{x}_i$ into the parameters of the corresponding conditional** $q(\mathbf{z}_i|\mathbf{x}_i) = \Psi(\mathbf{z}_i|g_\gamma(\mathbf{x}_i))$.

The family $(\Psi(\cdot|\kappa))_{\kappa\in\mathcal{K}}$ is a parametric family of distributions over $\mathbb{R}^d$ (usually Gaussians), $g_\gamma : \mathcal{X} \to K$ is a neural net called the **inference network.**

Inference for DLVMs solves the optimisation problem

$$\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta},\boldsymbol{\gamma}\in\boldsymbol{\Gamma}} \mathcal{L}(\boldsymbol{\theta}, Q_\gamma) = \max_{\boldsymbol{\theta}\in\boldsymbol{\Theta},\boldsymbol{\gamma}\in\boldsymbol{\Gamma}} \mathbb{E}_{\mathbf{Z}\sim Q_\gamma}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{X},\mathbf{Z})}{Q_\gamma(\mathbf{Z})}\right],$$

typically using **stochastic gradient descent.**[1]



A DLVM combined with AVI is called a **variational autoencoder (VAE).**

[1] Stochastic gradients of $\mathcal{L}(\boldsymbol{\theta}, Q_\gamma)$ can be calculated using the reparameterization trick.

Amortised Variational Inference (AVI)

Kingma and Welling (2014) and Rezende et al. (2014)

Amortised inference scales up VI by

(a) Writing Q as a product of conditional variational distributions:

$$Q_\gamma(\mathbf{Z}) = \prod_{i=1}^{n} q_\gamma(\mathbf{z}_i|\mathbf{x}_i)$$

(b) Learning a function g_γ that **transforms each data point $\mathbf{x}_i$ into the parameters of the corresponding conditional** $q(\mathbf{z}_i|\mathbf{x}_i) = \Psi(\mathbf{z}_i|g_\gamma(\mathbf{x}_i))$.

The family $(\Psi(\cdot|\kappa))_{\kappa\in\mathcal{K}}$ is a parametric family of distributions over $\mathbb{R}^d$ (usually

d the **inference network.**

Variational distribution

Note that:

- We can consider the variational distribtion to be an approxiamtion of the posterior $q_\gamma(\mathbf{z} \mid \mathbf{x}) \approx p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x})$
- The bound is tight, $\mathcal{L}(\boldsymbol{\theta}, Q_\gamma) = \ell(\boldsymbol{\theta})$, when $q_\gamma(\mathbf{z} \mid \mathbf{x}) = p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x})$

In

blem

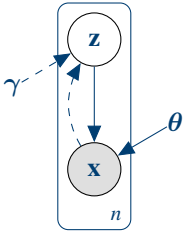
g $\dfrac{p_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{Z})}{Q_\gamma(\mathbf{Z})}\right]$,

ty

A

al autoencoder (VAE).

[1] Stochastic gradients of $\mathcal{L}(\boldsymbol{\theta}, Q_\gamma)$ can be calculated using the reparameterization trick.



Overview of talk

Introduction to Deep Latent Variable Models (DLVM)

Contribution 1: On the boundedness of the likelihood of DLVMs

Contribution 2: Missing data imputation using the exact conditional distribution

Contribution 3: Training and imputation on incomplete data sets

Was it sensible to maximise the likelihood in the first place?

If we see the prior as a mixing distribution, **DLVMs are continuous mixtures of distribution from the observation model:**

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int_{\mathbb{R}^d} p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z}) \, \mathrm{d}\mathbf{z} = \int_{\mathbb{R}^d} \Phi(\mathbf{x} | f_{\boldsymbol{\theta}}(\mathbf{z})) p(\mathbf{z}) \, \mathrm{d}\mathbf{z}.$$

But maximum likelihood for **finite Gaussian mixtures** is **ill-posed**:

- the likelihood function is unbounded (Day, 1969) and
- the parameters with infinite likelihood are pretty terrible.

Hence the question: **Is maximum likelihood well-posed for DLVMs?**

On the boundedness of the likelihood of DLVMs

Mattei and Frellsen (2018)

Consider a DLVM with a p-**variate Gaussian observation model** where

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log \int_{\mathbb{R}^d} \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}), \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{z})) p(\mathbf{z}) \, \mathrm{d}\mathbf{z}.$$

Like Kingma and Welling (2014), consider a **MLP decoder** with $h \in \mathbb{N}^*$ **hidden units** of the form

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}) = \mathbf{V} \tanh(\mathbf{W}\mathbf{z} + \mathbf{a}) + \mathbf{b}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{z}) = \exp(\boldsymbol{\alpha}^\intercal \tanh(\mathbf{W}\mathbf{z} + \mathbf{a}) + \beta)\mathbf{I}_p$$

where $\boldsymbol{\theta} = (\mathbf{W}, \mathbf{a}, \mathbf{V}, \mathbf{b}, \boldsymbol{\alpha}, \beta)$.

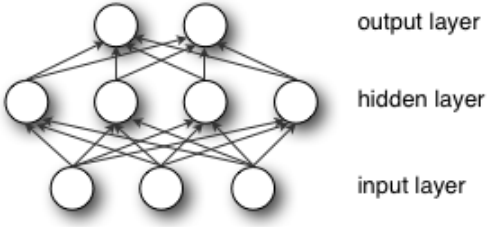


Image from http://deeplearning.net/tutorial/mlp.html

On the boundedness of the likelihood of DLVMs

Mattei and Frellsen (2018)

Consider a DLVM with a **p-variate Gaussian observation model** where

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log \int_{\mathbb{R}^d} \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}), \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{z})) p(\mathbf{z}) \, \mathrm{d}\mathbf{z}.$$

Like Kingma and Welling (2014), consider a **MLP decoder** with $h \in \mathbb{N}^*$ **hidden units** of the form

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}) = \mathbf{V} \tanh(\mathbf{W}\mathbf{z} + \mathbf{a}) + \mathbf{b}$$
$$\boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{z}) = \exp(\boldsymbol{\alpha}^\intercal \tanh(\mathbf{W}\mathbf{z} + \mathbf{a}) + \beta)\mathbf{I}_p$$

where $\boldsymbol{\theta} = (\mathbf{W}, \mathbf{a}, \mathbf{V}, \mathbf{b}, \boldsymbol{\alpha}, \beta)$.

Now consider a subfamily with $h = 1$ and

$$\boldsymbol{\theta}_k^{(i,\mathbf{w})} = (\alpha_k \mathbf{w}^\intercal, 0, 0, \mathbf{x}_{i*}, \alpha_k, -\alpha_k),$$

where $(\alpha_k)_{k \geq 1}$ is a nonnegative real sequence

$$\alpha_k \to \infty \text{ as } k \to \infty.$$

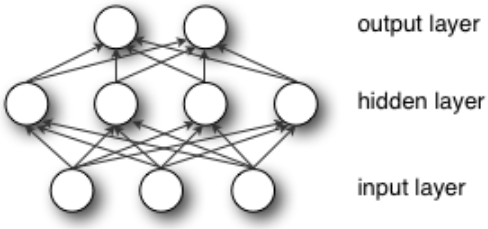


Image from http://deeplearning.net/tutorial/mlp.html

On the boundedness of the likelihood of DLVMs

Mattei and Frellsen (2018)

Consider a DLVM with a **p-variate Gaussian observation model** where

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log \int_{\mathbb{R}^d} \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}), \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{z})) p(\mathbf{z}) \, \mathrm{d}\mathbf{z}.$$

Like Kingma and Welling (2014), consider a **MLP decoder** with $h \in \mathbb{N}^*$ **hidden units** of the form

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}) = \mathbf{V} \tanh(\mathbf{W}\mathbf{z} + \mathbf{a}) + \mathbf{b} \qquad \boldsymbol{\mu}_{\boldsymbol{\theta}_k^{(i,\mathbf{w})}}(\mathbf{z}) = \mathbf{x}_{i*}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{z}) = \exp(\boldsymbol{\alpha}^\intercal \tanh(\mathbf{W}\mathbf{z} + \mathbf{a}) + \beta)\mathbf{I}_p \qquad \boldsymbol{\Sigma}_{\boldsymbol{\theta}_k^{(i,\mathbf{w})}}(\mathbf{z}) = \exp(\alpha_k \tanh(\alpha_k \mathbf{w}^\intercal \mathbf{z}) - \alpha_k)\mathbf{I}_p$$

where $\boldsymbol{\theta} = (\mathbf{W}, \mathbf{a}, \mathbf{V}, \mathbf{b}, \boldsymbol{\alpha}, \beta)$.

Now consider a subfamily with $h = 1$ and

$$\boldsymbol{\theta}_k^{(i,\mathbf{w})} = (\alpha_k \mathbf{w}^\intercal, 0, 0, \mathbf{x}_{i*}, \alpha_k, -\alpha_k),$$

where $(\alpha_k)_{k \geq 1}$ is a nonnegative real sequence

$$\alpha_k \to \infty \text{ as } k \to \infty.$$

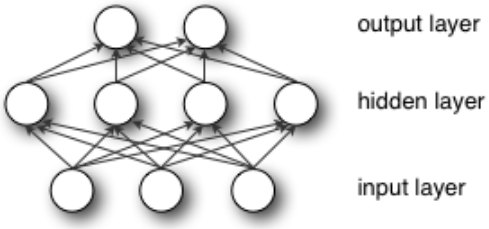


Image from http://deeplearning.net/tutorial/mlp.html

ML is ill-posed for a general Gaussian DLVM

Mattei and Frellsen (2018)

Theorem

For all $i^* \in \{1, \dots, n\}$ *and* $\mathbf{w} \in \mathbb{R}^d \setminus \{0\}$*, we have that* $\lim_{k \to \infty} \ell \left(\boldsymbol{\theta}_k^{(i^*, \mathbf{w})} \right) = \infty$.

Proof main idea: the contribution $\log p_{\boldsymbol{\theta}_k^{(i,\mathbf{w})}}(\mathbf{x}_{i^*})$ of the i^*-th observation explodes while all other contributions remain bounded below.

ML is ill-posed for a general Gaussian DLVM

Mattei and Frellsen (2018)

Theorem

For all $i^* \in \{1, \dots, n\}$ *and* $\mathbf{w} \in \mathbb{R}^d \setminus \{0\}$*, we have that* $\lim_{k \to \infty} \ell\left(\boldsymbol{\theta}_k^{(i^*, \mathbf{w})}\right) = \infty$.

Proof main idea: the contribution $\log p_{\boldsymbol{\theta}_k^{(i,\mathbf{w})}}(\mathbf{x}_{i^*})$ of the i^*-th observation explodes while all other contributions remain bounded below.

Do these infinite suprema lead to useful generative models?

Proposition

For all $k \in \mathbb{N}^*$, $i^* \in \{1, \dots, n\}$ *and* $\mathbf{w} \in \mathbb{R}^d \setminus \{0\}$*, the distribution* $p_{\boldsymbol{\theta}_k^{(i^*,\mathbf{w})}}(\mathbf{x})$ *is is spherically symmetric and unimodal around* $\mathbf{x}_{i^*}$.

No, because of the constant mean function.

ML is ill-posed for a general Gaussian DLVM

Mattei and Frellsen (2018)

Theorem

For all $i^* \in \{1, \dots, n\}$ *and* $\mathbf{w} \in \mathbb{R}^d \setminus \{0\}$*, we have that* $\lim_{k \to \infty} \ell\left(\boldsymbol{\theta}_k^{(i^*, \mathbf{w})}\right) = \infty$.

Proof main idea: the contribution $\log p_{\boldsymbol{\theta}_k^{(i, \mathbf{w})}}(\mathbf{x}_{i^*})$ of the i^*-th observation explodes while all other contributions remain bounded below.

Do these infinite suprema lead to useful generative models?

Proposition

For all $k \in \mathbb{N}^*$, $i^* \in \{1, \dots, n\}$ *and* $\mathbf{w} \in \mathbb{R}^d \setminus \{0\}$*, the distribution* $p_{\boldsymbol{\theta}_k^{(i^*, \mathbf{w})}}(\mathbf{x})$ *is is spherically symmetric and unimodal around* $\mathbf{x}_{i^*}$.

No, because of the constant mean function.

What about other parametrisations?

- The used MLP is a subfamily.
- Universal approximation abilities of neural networks.

ML is ill-posed for a general Gaussian DLVM

Mattei and Frellsen (2018)

Theorem

For all $i^* \in \{1, \dots, n\}$ *and* $\mathbf{w} \in \mathbb{R}^d \setminus \{0\}$*, we have that* $\lim_{k \to \infty} \ell\left(\boldsymbol{\theta}_k^{(i^*, \mathbf{w})}\right) = \infty$.

Proof main idea: the contribution $\log p_{\boldsymbol{\theta}_k^{(i,\mathbf{w})}}(\mathbf{x}_{i^*})$ of the i^*-th observation explodes while all other contributions remain bounded below.

Do these infinite suprema lead to useful generative models?

Proposition

For all $k \in \mathbb{N}^*$, $i^* \in \{1, \dots, n\}$ *and* $\mathbf{w} \in \mathbb{R}^d \setminus \{0\}$*, the distribution* $p_{\boldsymbol{\theta}_k^{(i^*,\mathbf{w})}}(\mathbf{x})$ *is is spherically symmetric and unimodal around* $\mathbf{x}_{i^*}$.

No, because of the constant mean function.

What about other parametrisations?

- The used MLP is a subfamily.
- Universal approximation abilities of neural networks.

What about discrete DLVM?

It is easy to show that **discrete DLVMs do not suffer from unbounded likelihood**.

Tackling the unboundedness of the likelihood

(Mattei and Frellsen, 2018)

Proposition

Let $\xi > 0$. If the parametrisation of the decoder is such that the image of $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ is included in

$$S_p^{\xi} = \{\mathbf{A} \in S_p^{+} \,|\, \min(\mathrm{Sp}\,\mathbf{A}) \geq \xi\}$$

for all $\boldsymbol{\theta}$, then the log-likelihood function is upper bounded by $-np \log \sqrt{\pi\xi}$

Note: Such constraints can be implemented by added $\xi \mathbf{I}_p$ to $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{z})$.

A data-dependent likelihood upper bounds

Mattei and Frellsen (2018) and figure extend from Cremer et al. (2018)

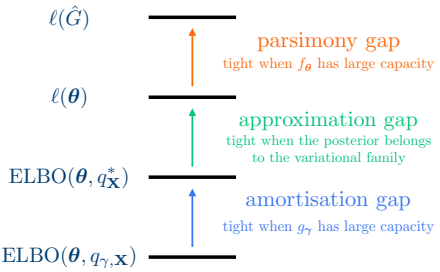
We can interpret DLVM as **parsimonious submodel** of **nonparametric mixture** model

$$p_G(\mathbf{x}) = \int_H \Psi(\mathbf{x}|\eta)\,\mathrm{d}G(\eta) \qquad \ell(G) = \sum_{i=1}^{n} \log p_G(\mathbf{x}_i).$$

- The model parameter is the **mixing distribution** $G \in \mathcal{P}$, where $\mathcal{P}$ is the set of all probability measures over parameter space H.
- This is a DLVM, when G is generatively defined by: $\mathbf{z} \sim p(\mathbf{z}); \eta = f_{\boldsymbol{\theta}}(\mathbf{z})$.

This gives us an **immediate upper bound on the likelihood for any decoder** $f_{\boldsymbol{\theta}}$:

$$\ell(\boldsymbol{\theta}) \leq \max_{G \in \mathcal{P}} \ell(G)$$



A data-dependent likelihood upper bounds

Mattei and Frellsen (2018) and figure extend from Cremer et al. (2018)

We can interpret DLVM as **parsimonious submodel** of **nonparametric mixture** model

$$p_G(\mathbf{x}) = \int_H \Psi(\mathbf{x}|\eta)\,\mathrm{d}G(\eta) \qquad \ell(G) = \sum_{i=1}^{n} \log p_G(\mathbf{x}_i).$$

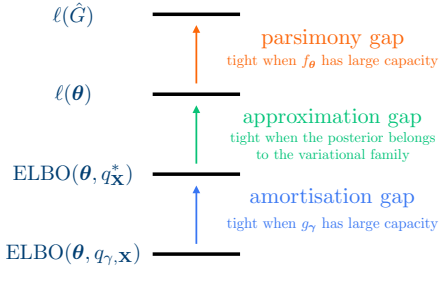
- The model parameter is the **mixing distribution** $G \in \mathcal{P}$, where $\mathcal{P}$ is the set of all probability measures over parameter space H.
- This is a DLVM, when G is generatively defined by: $\mathbf{z} \sim p(\mathbf{z}); \eta = f_{\boldsymbol{\theta}}(\mathbf{z})$.

This gives us an **immediate upper bound on the likelihood for any decoder** $f_{\boldsymbol{\theta}}$:

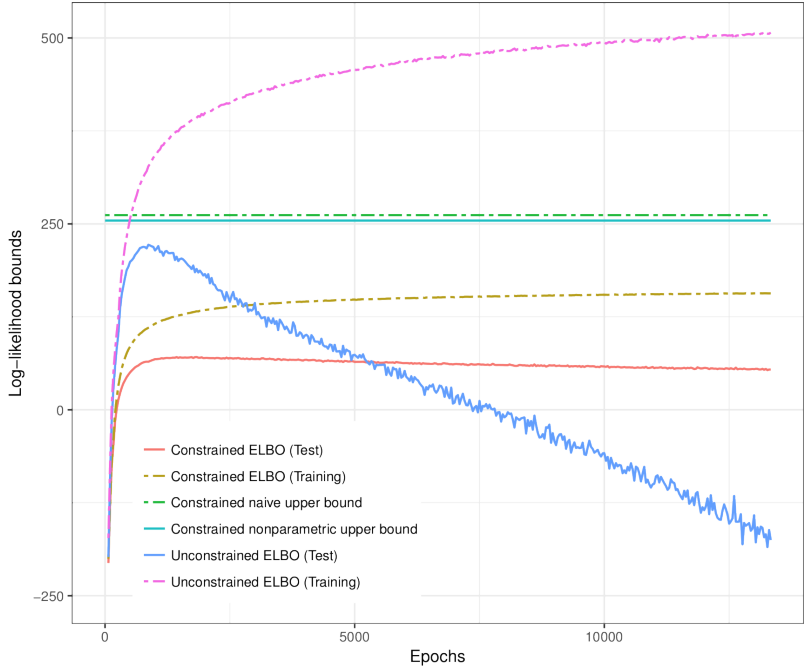
$$\ell(\boldsymbol{\theta}) \leq \max_{G \in \mathcal{P}} \ell(G)$$

Theorem

Assume that $(\Psi(\cdot \mid \eta))_{\eta \in H}$ is the family of multivariate Bernoulli distributions or Gaussian distributions with the spectral constraint. The likelihood of the nonparametric mixture model is maximised for a finite mixture model of $k \leq n$ distributions from the family $(\Psi(\cdot \mid \eta))_{\eta \in H}$.



Unboundedness for a DLVM with Gaussian outputs (Frey faces)



Overview of talk

Introduction to Deep Latent Variable Models (DLVM)

Contribution 1: On the boundedness of the likelihood of DLVMs

Contribution 2: Missing data imputation using the exact conditional distribution

Contribution 3: Training and imputation on incomplete data sets

Data imputation with variational autoencoders

Mattei and Frellsen (2018)

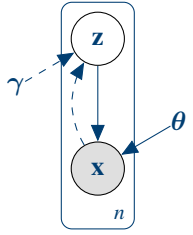
After training a couple encoder/decoder, we consider a **new data point** $\mathbf{x} = (\mathbf{x}^{\text{obs}}, \mathbf{x}^{\text{miss}})$.

In principle we can **impute** $\mathbf{x}^{\text{miss}}$ using

$$p_{\boldsymbol{\theta}}(\mathbf{x}^{\text{miss}} \mid \mathbf{x}^{\text{obs}}) = \int_{\mathbb{R}^d} \Phi(\mathbf{x}^{\text{miss}} \mid \mathbf{x}^{\text{obs}}, f_{\boldsymbol{\theta}}(\mathbf{z})) p(\mathbf{z} | \mathbf{x}^{\text{obs}}) \, \mathrm{d}\mathbf{z}.$$

Since this integral is intractable, Rezende et al. (2014) suggested using **pseudo-Gibbs sampling**, by forming a Markov chain $(\mathbf{z}_t, \hat{\mathbf{x}}_t^{\text{miss}})_{t \geq 1}$

- $\mathbf{z}_t \sim \Psi(\mathbf{z} \mid g_{\boldsymbol{\gamma}}(\mathbf{x}^{\text{obs}}, \hat{\mathbf{x}}_{t-1}^{\text{miss}}))$
- $\hat{\mathbf{x}}_t^{\text{miss}} \sim \Phi(\mathbf{x}^{\text{miss}} \mid \mathbf{x}^{\text{obs}}, f_{\boldsymbol{\theta}}(\mathbf{z}_t))$



Data imputation with variational autoencoders

Mattei and Frellsen (2018)

After training a couple encoder/decoder, we consider a **new data point** $\mathbf{x} = (\mathbf{x}^{\text{obs}}, \mathbf{x}^{\text{miss}})$.

In principle we can **impute** $\mathbf{x}^{\text{miss}}$ using

$$p_{\boldsymbol{\theta}}(\mathbf{x}^{\text{miss}} \mid \mathbf{x}^{\text{obs}}) = \int_{\mathbb{R}^d} \Phi(\mathbf{x}^{\text{miss}} \mid \mathbf{x}^{\text{obs}}, f_{\boldsymbol{\theta}}(\mathbf{z})) p(\mathbf{z} | \mathbf{x}^{\text{obs}}) \, \mathrm{d}\mathbf{z}.$$

Since this integral is intractable, Rezende et al. (2014) suggested using **pseudo-Gibbs sampling**, by forming a Markov chain $(\mathbf{z}_t, \hat{\mathbf{x}}_t^{\text{miss}})_{t \geq 1}$

- $\mathbf{z}_t \sim \Psi(\mathbf{z} \mid g_{\boldsymbol{\gamma}}(\mathbf{x}^{\text{obs}}, \hat{\mathbf{x}}_{t-1}^{\text{miss}}))$
- $\hat{\mathbf{x}}_t^{\text{miss}} \sim \Phi(\mathbf{x}^{\text{miss}} \mid \mathbf{x}^{\text{obs}}, f_{\boldsymbol{\theta}}(\mathbf{z}_t))$

We propose Metropolis-within-Gibbs:

Algorithm 1 Metropolis-within-Gibbs sampler for missing data imputation using a trained VAE

Inputs: Observed data $\mathbf{x}^{\text{obs}}$, trained VAE $(f_{\boldsymbol{\theta}}, g_{\boldsymbol{\gamma}})$, number of iterations T
Outputs: Markov chain of imputations $\hat{\mathbf{x}}_1^{\text{miss}}, ..., \hat{\mathbf{x}}_T^{\text{miss}}$.
Initialise $(\mathbf{z}_0, \hat{\mathbf{x}}_0^{\text{miss}})$
for $t = 1$ **to** T **do**

$$\tilde{\mathbf{z}}_t \sim \Psi(\mathbf{z} | g_{\boldsymbol{\gamma}}(\mathbf{x}^{\text{obs}}, \hat{\mathbf{x}}_{t-1}^{\text{miss}}))$$

$$\tilde{\rho}_t = \frac{\Phi(\mathbf{x}^{\text{obs}}, \hat{\mathbf{x}}_{t-1}^{\text{miss}} | f_{\boldsymbol{\theta}}(\tilde{\mathbf{z}}_t)) p(\tilde{\mathbf{z}}_t)}{\Phi(\mathbf{x}^{\text{obs}}, \hat{\mathbf{x}}_{t-1}^{\text{miss}} | f_{\boldsymbol{\theta}}(\mathbf{z}_{t-1})) p(\mathbf{z}_{t-1})} \frac{\Psi(\mathbf{z}_{t-1} | g_{\boldsymbol{\gamma}}(\mathbf{x}^{\text{obs}}, \hat{\mathbf{x}}_{t-1}^{\text{miss}}))}{\Psi(\tilde{\mathbf{z}}_t | g_{\boldsymbol{\gamma}}(\mathbf{x}^{\text{obs}}, \hat{\mathbf{x}}_{t-1}^{\text{miss}}))}$$

$$\rho_t = \min\{\tilde{\rho}_t, 1\}$$

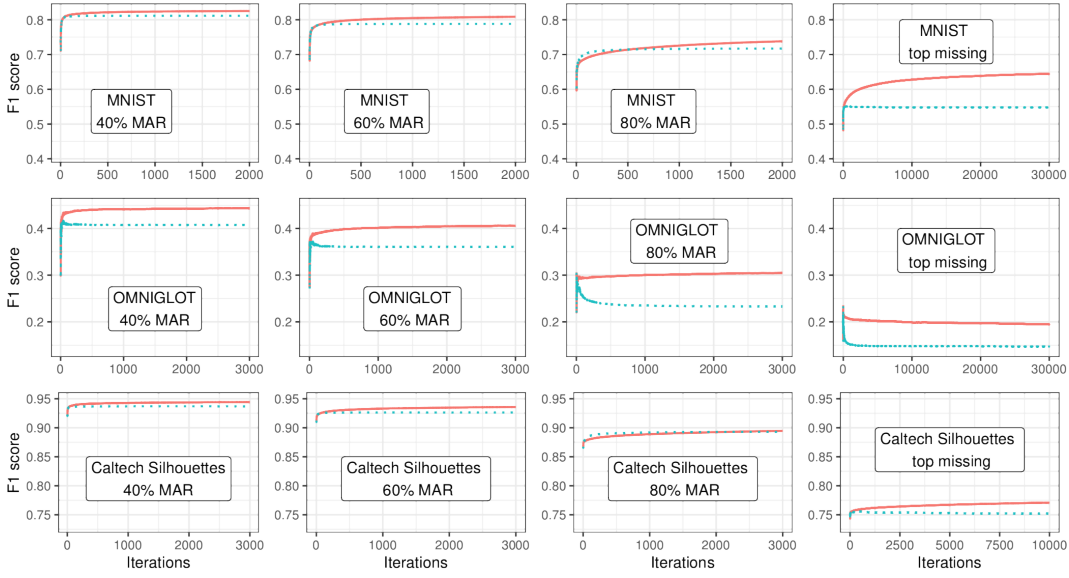
$$\mathbf{z}_t = \begin{cases} \tilde{\mathbf{z}}_t & \text{with probability } \rho_t \\ \mathbf{z}_{t-1} & \text{with probability } 1 - \rho_t \end{cases}$$

$$\hat{\mathbf{x}}_t^{\text{miss}} \sim p_{\boldsymbol{\theta}}(\mathbf{x}^{\text{miss}} | \mathbf{x}^{\text{obs}}, \mathbf{z}_t)$$

end for

Comparing pseudo-Gibbs and Metropolis-within-Gibbs

Mattei and Frellsen (2018)



The F1 score is the harmonic mean of the precision and recall, $F_1 = ((\text{recall}^{-1} + \text{precision}^{-1})/2)^{-1}$.

Overview of talk

Introduction to Deep Latent Variable Models (DLVM)

Contribution 1: On the boundedness of the likelihood of DLVMs

Contribution 2: Missing data imputation using the exact conditional distribution

Contribution 3: Training and imputation on incomplete data sets

What happens if the training set is incomplete?

Assume that some of the training and test data is **missing-at-random** (MAR).

We can then split each sample $i \in \{1, \ldots, n\}$ into

- the **observed features** $\mathbf{x}_i^{\mathrm{o}}$ and
- the **missing features** $\mathbf{x}_i^{\mathrm{m}}$.

Under the MAR assumption, the relevant quantity to **maximise is the likelihood of the observed data** equal to

$$\ell^{\mathrm{o}}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}(\mathbf{x}_i^{\mathrm{o}}) = \sum_{i=1}^{n} \log \int p_{\boldsymbol{\theta}}(\mathbf{x}_i^{\mathrm{o}} \mid \mathbf{z}) p(\mathbf{z}) \, \mathrm{d}\mathbf{z}.$$

Direct MLE is intractable, but we can derive tractable tight lower bounds of $\ell(\boldsymbol{\theta})$.

The importance-weighted autoencoder (IWAE)

Burda et al. (2016)

Let us revisit the full data likelihood:

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}(\mathbf{x}_i) = \int_{\mathbb{R}^d} p_{\boldsymbol{\theta}}(\mathbf{x}_i \mid \mathbf{z}) p(\mathbf{z}) \, \mathrm{d}\mathbf{z}.$$

VAE bound

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\gamma}) = \sum_{i=1}^{n} \mathbb{E}_{\mathbf{z} \sim q_{\boldsymbol{\gamma}}(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_i \mid \mathbf{z}_i) p(\mathbf{z}_i)}{q_{\boldsymbol{\gamma}}(\mathbf{z}_i|\mathbf{x}_i)} \right] = \ell(\boldsymbol{\theta}) - \mathrm{KL}(q_{\boldsymbol{\gamma}}(\mathbf{Z} \mid \mathbf{X}) \,||\, p_{\boldsymbol{\theta}}(\mathbf{Z} \mid \mathbf{X})) \leq \ell(\boldsymbol{\theta})$$

The importance-weighted autoencoder (IWAE)

Burda et al. (2016)

Let us revisit the full data likelihood:

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}(\mathbf{x}_i) = \int_{\mathbb{R}^d} p_{\boldsymbol{\theta}}(\mathbf{x}_i \mid \mathbf{z}) p(\mathbf{z}) \, \mathrm{d}\mathbf{z}.$$

VAE bound

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\gamma}) = \sum_{i=1}^{n} \mathbb{E}_{\mathbf{z} \sim q_{\boldsymbol{\gamma}}(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_i \mid \mathbf{z}_i) p(\mathbf{z}_i)}{q_{\boldsymbol{\gamma}}(\mathbf{z}_i|\mathbf{x}_i)} \right] = \ell(\boldsymbol{\theta}) - \mathrm{KL}(q_{\boldsymbol{\gamma}}(\mathbf{Z} \mid \mathbf{X}) \,||\, p_{\boldsymbol{\theta}}(\mathbf{Z} \mid \mathbf{X})) \leq \ell(\boldsymbol{\theta})$$

IWAE bound

We can obtain a *strictly tight bound* based on **importance sampling**

$$\mathcal{L}_K(\boldsymbol{\theta}, \boldsymbol{\gamma}) = \sum_{i=1}^{n} \mathbb{E}_{\mathbf{z}_{i1}, \ldots, \mathbf{z}_{iK} \sim q_{\boldsymbol{\gamma}}(\mathbf{z}|\mathbf{x}_i)} \left[\log \frac{1}{K} \sum_{k=1}^{K} \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}_{ik}) p(\mathbf{z}_{ik})}{q_{\boldsymbol{\gamma}}(\mathbf{z}_{ik}|\mathbf{x}_i)} \right].$$

Here $q_{\boldsymbol{\gamma}}(\mathbf{z}|\mathbf{x}_i)$ play the role of a **proposal distribution** close to the posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$, and

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\gamma}) = \mathcal{L}_1(\boldsymbol{\theta}, \boldsymbol{\gamma}) \leq \mathcal{L}_2(\boldsymbol{\theta}, \boldsymbol{\gamma}) \leq \ldots \leq \mathcal{L}_K(\boldsymbol{\theta}, \boldsymbol{\gamma}) \xrightarrow[K \to \infty]{} \ell(\boldsymbol{\theta}).$$

The missing data importance-weighted autoencoder (MIWAE) bound

Mattei and Frellsen (2019)

For the case of **missing data**, we propose the variational distribution

$$q_{\boldsymbol{\gamma}}(\mathbf{z}|\mathbf{x}^{\mathrm{o}}) = \Psi(\mathbf{z}|g_{\boldsymbol{\gamma}}(\iota(\mathbf{x}^{\mathrm{o}})),$$

where:

- The set $(\Psi(\cdot|\boldsymbol{\kappa}))_{\boldsymbol{\kappa}\in\mathcal{K}}$ is the *variational family*.
- The function $g_{\boldsymbol{\gamma}} : \mathcal{X} \to \mathcal{K}$ is the *encoder*.
- ι is an **imputation function chosen beforehand** that transforms $\mathbf{x}^{\mathrm{o}}$ into a complete input vector $\iota(\mathbf{x}^{\mathrm{o}}) \in \mathcal{X}$ such that $\iota(\mathbf{x}^{\mathrm{o}})^{\mathrm{o}} = \mathbf{x}^{\mathrm{o}}$.

The missing data importance-weighted autoencoder (MIWAE) bound

Mattei and Frellsen (2019)

For the case of **missing data**, we propose the variational distribution

$$q_{\boldsymbol{\gamma}}(\mathbf{z}|\mathbf{x}^{\mathrm{o}}) = \Psi(\mathbf{z}|g_{\boldsymbol{\gamma}}(\iota(\mathbf{x}^{\mathrm{o}})),$$

where:

- The set $(\Psi(\cdot|\boldsymbol{\kappa}))_{\boldsymbol{\kappa}\in\mathcal{K}}$ is the *variational family*.
- The function $g_{\boldsymbol{\gamma}} : \mathcal{X} \to \mathcal{K}$ is the *encoder*.
- ι is an **imputation function chosen beforehand** that transforms $\mathbf{x}^{\mathrm{o}}$ into a complete input vector $\iota(\mathbf{x}^{\mathrm{o}}) \in \mathcal{X}$ such that $\iota(\mathbf{x}^{\mathrm{o}})^{\mathrm{o}} = \mathbf{x}^{\mathrm{o}}$.

Following Burda et al. (2016), we can use the distribution $q_{\boldsymbol{\gamma}}$ to build lower bounds of $\ell^{\mathrm{o}}(\boldsymbol{\theta})$

$$\mathcal{L}_K^{\mathrm{o}}(\boldsymbol{\theta},\boldsymbol{\gamma}) = \sum_{i=1}^{n} \mathbb{E}_{\mathbf{z}_{i1},\ldots,\mathbf{z}_{iK}\sim q_{\boldsymbol{\gamma}}(\mathbf{z}|\mathbf{x}_i^{\mathrm{o}})} \left[\log \frac{1}{K} \sum_{k=1}^{K} \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_i^{\mathrm{o}}|\mathbf{z}_{ik}) p(\mathbf{z}_{ik})}{q_{\boldsymbol{\gamma}}(\mathbf{z}_{ik}|\mathbf{x}_i^{\mathrm{o}})} \right].$$

and show that

$$\mathcal{L}_1^{\mathrm{o}}(\boldsymbol{\theta},\boldsymbol{\gamma}) \leq \mathcal{L}_2^{\mathrm{o}}(\boldsymbol{\theta},\boldsymbol{\gamma}) \leq \ldots \leq \mathcal{L}_K^{\mathrm{o}}(\boldsymbol{\theta},\boldsymbol{\gamma}) \xrightarrow[K\to\infty]{} \ell(\boldsymbol{\theta}).$$

The missing data importance-weighted autoencoder (MIWAE) bound

Mattei and Frellsen (2019)

Imputation function

When $K \to \infty$, the bound is tight for any imputation function ι

We use zero-imputation.

ne variational distribution

$\Psi(\mathbf{z}|g_{\boldsymbol{\gamma}}(\iota(\mathbf{x}^{\mathrm{o}})),$

mily.

- The function $g_{\boldsymbol{\gamma}} : \mathcal{X} \to \mathcal{K}$ is the *encoder*.
- ι is an **imputation function chosen beforehand** that transforms $\mathbf{x}^{\mathrm{o}}$ into a complete input vector $\iota(\mathbf{x}^{\mathrm{o}}) \in \mathcal{X}$ such that $\iota(\mathbf{x}^{\mathrm{o}})^{\mathrm{o}} = \mathbf{x}^{\mathrm{o}}$.

Following Burda et al. (2016), we can use the distribution $q_{\boldsymbol{\gamma}}$ to build lower bounds of $\ell^{\mathrm{o}}(\boldsymbol{\theta})$

$$\mathcal{L}_K^{\mathrm{o}}(\boldsymbol{\theta}, \boldsymbol{\gamma}) = \sum_{i=1}^{n} \mathbb{E}_{\mathbf{z}_{i1},\ldots,\mathbf{z}_{iK} \sim q_{\boldsymbol{\gamma}}(\mathbf{z}|\mathbf{x}_i^{\mathrm{o}})} \left[\log \frac{1}{K} \sum_{k=1}^{K} \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_i^{\mathrm{o}}|\mathbf{z}_{ik}) p(\mathbf{z}_{ik})}{q_{\boldsymbol{\gamma}}(\mathbf{z}_{ik}|\mathbf{x}_i^{\mathrm{o}})} \right].$$

and show that

$$\mathcal{L}_1^{\mathrm{o}}(\boldsymbol{\theta}, \boldsymbol{\gamma}) \leq \mathcal{L}_2^{\mathrm{o}}(\boldsymbol{\theta}, \boldsymbol{\gamma}) \leq \ldots \leq \mathcal{L}_K^{\mathrm{o}}(\boldsymbol{\theta}, \boldsymbol{\gamma}) \xrightarrow[K \to \infty]{} \ell(\boldsymbol{\theta}).$$

The missing data importance-weighted autoencoder (MIWAE) bound

Mattei and Frellsen (2019)

Imputation function

When $K \to \infty$, the bound is tight for any imputation function ι

We use zero-imputation.

MVAE bound

When $K = 1$, the bound resembles the VAE bound and we call it MVAE.

This bound was independently derived independently by Nazabal et al. (2018) in concurrent work.

- The function $g_\gamma : \mathcal{X} \to \mathcal{K}$ is the *encoder*.
- ι is an **imputation function chosen beforehand** that transforms $\mathbf{x}^{\mathrm{o}}$ into a complete input vector $\iota(\mathbf{x}^{\mathrm{o}}) \in \mathcal{X}$ such that $\iota(\mathbf{x}^{\mathrm{o}})^{\mathrm{o}} = \mathbf{x}^{\mathrm{o}}$.

Following Burda et al. (2016), we can use the distribution q_γ to build lower bounds of $\ell^{\mathrm{o}}(\boldsymbol{\theta})$

$$\mathcal{L}_K^{\mathrm{o}}(\boldsymbol{\theta}, \boldsymbol{\gamma}) = \sum_{i=1}^{n} \mathbb{E}_{\mathbf{z}_{i1}, \ldots, \mathbf{z}_{iK} \sim q_{\boldsymbol{\gamma}}(\mathbf{z}|\mathbf{x}_i^{\mathrm{o}})} \left[\log \frac{1}{K} \sum_{k=1}^{K} \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_i^{\mathrm{o}}|\mathbf{z}_{ik}) p(\mathbf{z}_{ik})}{q_{\boldsymbol{\gamma}}(\mathbf{z}_{ik}|\mathbf{x}_i^{\mathrm{o}})} \right].$$

and show that

$$\mathcal{L}_1^{\mathrm{o}}(\boldsymbol{\theta}, \boldsymbol{\gamma}) \leq \mathcal{L}_2^{\mathrm{o}}(\boldsymbol{\theta}, \boldsymbol{\gamma}) \leq \ldots \leq \mathcal{L}_K^{\mathrm{o}}(\boldsymbol{\theta}, \boldsymbol{\gamma}) \xrightarrow[K \to \infty]{} \ell(\boldsymbol{\theta}).$$

Imputation with MIWAE

Mattei and Frellsen (2019)

For the **single imputation problem** the optimal decision-theoretic choice is

$$\hat{\mathbf{x}}^{\mathrm{m}} = \mathbb{E}[\mathbf{x}^{\mathrm{m}}|\mathbf{x}^{\mathrm{o}}] = \int \mathbf{x}^{\mathrm{m}} p_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{m}}|\mathbf{x}^{\mathrm{o}})\, \mathrm{d}\mathbf{x}^{\mathrm{m}} = \iint \mathbf{x}^{\mathrm{m}} p_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{m}}|\mathbf{x}^{\mathrm{o}}, \mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{\mathrm{o}})\, \mathrm{d}\mathbf{z} d\mathbf{x}^{\mathrm{m}},$$

Imputation with MIWAE

Mattei and Frellsen (2019)

For the **single imputation problem** the optimal decision-theoretic choice is

$$\hat{\mathbf{x}}^{\mathrm{m}} = \mathbb{E}[\mathbf{x}^{\mathrm{m}}|\mathbf{x}^{\mathrm{o}}] = \int \mathbf{x}^{\mathrm{m}} p_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{m}}|\mathbf{x}^{\mathrm{o}})\, \mathrm{d}\mathbf{x}^{\mathrm{m}} = \iint \mathbf{x}^{\mathrm{m}} p_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{m}}|\mathbf{x}^{\mathrm{o}}, \mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{\mathrm{o}})\, \mathrm{d}\mathbf{z} d\mathbf{x}^{\mathrm{m}},$$

This is intractable, but can be estimated using **self-normalised importance sampling** with the proposal distribution $p_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{m}}|\mathbf{x}^{\mathrm{o}}, \mathbf{z}) q_{\boldsymbol{\gamma}}(\mathbf{z}|\mathbf{x}^{\mathrm{o}})$, leading to the estimate

$$\mathbb{E}[\mathbf{x}^{\mathrm{m}}|\mathbf{x}^{\mathrm{o}}] \approx \sum_{l=1}^{L} w_l\, \mathbf{x}^{\mathrm{m}}_{(l)},$$

where $(\mathbf{x}^{\mathrm{m}}_{(1)}, \mathbf{z}_{(1)}), \ldots, (\mathbf{x}^{\mathrm{m}}_{(L)}, \mathbf{z}_{(L)})$ are i.i.d. samples from $p_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{m}}|\mathbf{x}^{\mathrm{o}}, \mathbf{z}) q_{\boldsymbol{\gamma}}(\mathbf{z}|\mathbf{x}^{\mathrm{o}})$ and

$$w_l = \frac{r_l}{r_1 + \ldots + r_L}, \text{ with } r_l = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{o}}|\mathbf{z}_{(l)}) p(\mathbf{z}_{(l)})}{q_{\boldsymbol{\gamma}}(\mathbf{z}_{(l)}|\mathbf{x}^{\mathrm{o}})}.$$

Here, we leverage the fact that $q_{\boldsymbol{\gamma}}(\mathbf{z}|\mathbf{x}^{\mathrm{o}})$ is a good approximation of $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{\mathrm{o}})$.

Imputation with MIWAE

Mattei and Frellsen (2019)

For the **single imputation problem** the optimal decision-theoretic choice is

$$\hat{\mathbf{x}}^{\mathrm{m}} = \mathbb{E}[\mathbf{x}^{\mathrm{m}}|\mathbf{x}^{\mathrm{o}}] = \int \mathbf{x}^{\mathrm{m}} p_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{m}}|\mathbf{x}^{\mathrm{o}})\, \mathrm{d}\mathbf{x}^{\mathrm{m}} = \iint \mathbf{x}^{\mathrm{m}} p_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{m}}|\mathbf{x}^{\mathrm{o}}, \mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{\mathrm{o}})\, \mathrm{d}\mathbf{z} d\mathbf{x}^{\mathrm{m}},$$

This is intractable, but can be estimated using **self-normalised importance sampling** with the proposal distribution $p_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{m}}|\mathbf{x}^{\mathrm{o}}, \mathbf{z}) q_{\boldsymbol{\gamma}}(\mathbf{z}|\mathbf{x}^{\mathrm{o}})$, leading to the estimate

$$\mathbb{E}[\mathbf{x}^{\mathrm{m}}|\mathbf{x}^{\mathrm{o}}] \approx \sum_{l=1}^{L} w_l\, \mathbf{x}^{\mathrm{m}}_{(l)},$$

where $(\mathbf{x}^{\mathrm{m}}_{(1)}, \mathbf{z}_{(1)}), \ldots, (\mathbf{x}^{\mathrm{m}}_{(L)}, \mathbf{z}_{(L)})$ are i.i.d. samples from $p_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{m}}|\mathbf{x}^{\mathrm{o}}, \mathbf{z}) q_{\boldsymbol{\gamma}}(\mathbf{z}|\mathbf{x}^{\mathrm{o}})$ and

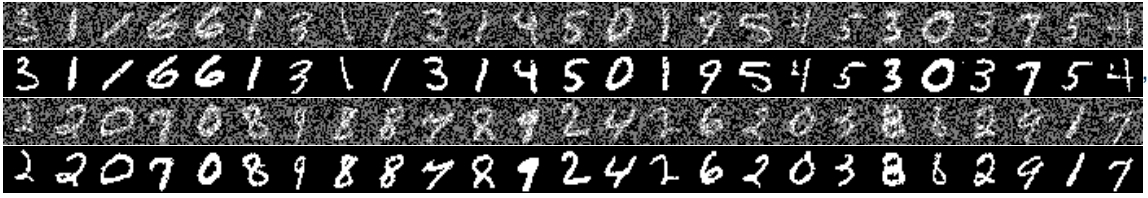
$$w_l = \frac{r_l}{r_1 + \ldots + r_L}, \text{ with } r_l = \frac{p_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{o}}|\mathbf{z}_{(l)}) p(\mathbf{z}_{(l)})}{q_{\boldsymbol{\gamma}}(\mathbf{z}_{(l)}|\mathbf{x}^{\mathrm{o}})}.$$

Here, we leverage the fact that $q_{\boldsymbol{\gamma}}(\mathbf{z}|\mathbf{x}^{\mathrm{o}})$ is a good approximation of $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{\mathrm{o}})$.

Multiple imputation, i.e. sampling from $p_{\boldsymbol{\theta}}(\mathbf{x}^{\mathrm{m}}|\mathbf{x}^{\mathrm{o}})$, can be done using **sampling importance resampling** according to the weights w_l for large L.

Convolutional MIWAE on binary MNIST (50% MAR pixels)

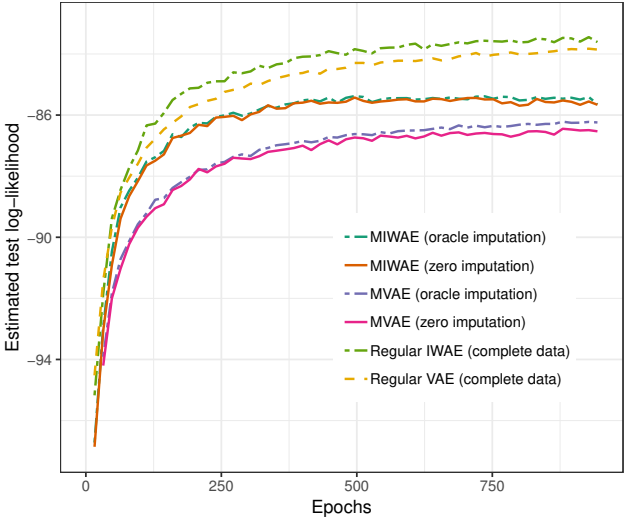
Mattei and Frellsen (2019)



Random incomplete samples from the MNIST training data set, and the imputations obtained by MIWAE (trained with $K = 50$ importance weights, and imputed with $L = 10\,000$ importance weights)

Convolutional MIWAE on binary MNIST (50% MAR pixels)

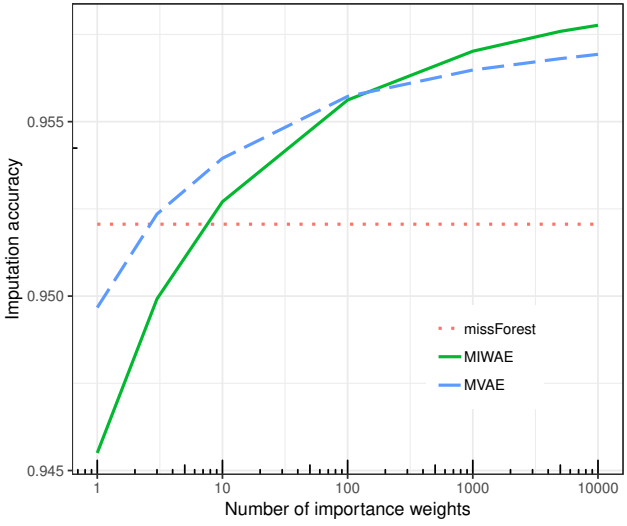
Mattei and Frellsen (2019)



Estimated test log-likelihood of various models trained on binary MNIST as a function of the number of training epochs. The MIWAE model was trained using $K = 50$ importance weights.

Convolutional MIWAE on binary MNIST (50% MAR pixels)

Mattei and Frellsen (2019)



Test imputation accuracy as a function of the number of importance weights (L) used in the **single imputation scheme**. The MIWAE model was trained using $K = 50$ importance weights.

Multiple imputation on binary MNIST (50% MAR pixels)

Mattei and Frellsen (2019)

To evaluate **multiple imputation**, we consider the task of classifying the incomplete binary MNIST data set.

	Test accuracy	*Test cross-entropy*
Zero imp.	0.9739 (0.0018)	0.1003 (0.0092)
missForest imp.	0.9805 (0.0018)	0.0645 (0.0066)
MIWAE single imp.	0.9847 (0.0009)	0.0510 (0.0035)
MIWAE multiple imp.	**0.9868 (0.0008)**	0.0509 (0.0044)
Complete data	0.9866 (0.0007)	**0.0464 (0.0026)**

Test accuracy and cross-entropy obtained by training a convolutional network using the imputed versions of the static binarisation of MNIST. The numbers are the mean of 10 repeated trainings with different seeds and standard deviations are shown in brackets.

Single imputation of UCI data sets (50% MAR)

Mattei and Frellsen (2019)

For all data sets **we train DLVMs with the same general properties**:

- Both encoder and decoder are multi-layer perceptrons with 3 hidden layers (128 hidden units) and tanh activations.
- Products of Student's t for both the variational family and the observation model
- Same number of gradient steps ($500\,000$) for all data sets, and no regularisation.

	Banknote	*Breast*	*Concrete*	*Red*	*White*	*Yeast*
MIWAE	**0.446 (0.038)**	**0.280 (0.021)**	**0.501 (0.040)**	**0.643 (0.026)**	**0.735 (0.033)**	**0.964(0.057)**
MVAE	0.593 (0.059)	0.318 (0.018)	0.587(0.026)	0.686 (0.120)	0.782 (0.018)	0.997 (0.064)
missForest	0.676 (0.040)	0.291 (0.026)	0.510 (0.11)	0.697 (0.050)	0.798 (0.019)	1.41 (0.02)
PCA	0.682 (0.016)	0.729 (0.068)	0.938 (0.033)	0.890 (0.033)	0.865 (0.024)	1.05(0.061)
kNN	0.744 (0.033)	0.831 (0.029)	0.962(0.034)	0.981 (0.037)	0.929 (0.025)	1.17 (0.048)
Mean	1.02 (0.032)	1.00 (0.04)	1.01 (0.035)	1.00 (0.03)	1.00 (0.02)	1.06 (0.052)

Mean-squared error for single imputation for various continuous UCI data sets (mean and standard deviations over 5 randomly generated incomplete data sets).

Summary

Take-home message

- DLVMs are flexible generative models.
- Showed that **MLE is ill-posed** for unconstrained DLVMs with Gaussian output.
- Propose how to **tackle this problem** using constraints.
- Provided an **upper bound for the likelihood** in well-posed cases.
- Showed how to **draw samples according to the exact conditional distribution** with missing data.
- Showed how to **train DLVM with missing data**
- Obtained **state-of-the-art performance in missing data imputation**

Summary

Take-home message

- DLVMs are flexible generative models.
- Showed that **MLE is ill-posed** for unconstrained DLVMs with Gaussian output.
- Propose how to **tackle this problem** using constraints.
- Provided an **upper bound for the likelihood** in well-posed cases.
- Showed how to **draw samples according to the exact conditional distribution** with missing data.
- Showed how to **train DLVM with missing data**
- Obtained **state-of-the-art performance in missing data imputation**

Thank you for your attention!

Questions?

References

- Blei, D. M., A. Kucukelbir, and J. D. McAuliffe (2017). “Variational Inference: A Review for Statisticians”. In: *Journal of the American Statistical Association* 112.518, pp. 859–877.
- Burda, Y., R. Grosse, and R. Salakhutdinov (2016). “Importance weighted autoencoders”. In: *Proceedings of the International Conference on Learning Representations*.
- Cremer, C., X. Li, and D. Duvenaud (2018). “Inference Suboptimality in Variational Autoencoders”. In: *arXiv:1801.03558*.
- Day, N. E. (1969). “Estimating the Components of a Mixture of Normal Distributions”. In: *Biometrika* 56.3, pp. 463–474.
- Harchaoui, W., P.-A. Mattei, A. Alamansa, and C. Bouveyron (2018). “Wasserstein Adversarial Mixture Clustering”.
- Kingma, D. P. and M. Welling (2014). “Auto-encoding variational Bayes”. In: *International Conference on Learning Representations*.
- Kusner, M. J., B. Paige, and J. M. Hernández-Lobato (2017). “Grammar Variational Autoencoder”. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by D. Precup and Y. W. Teh. Vol. 70. Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia: PMLR, pp. 1945–1954.
- MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge University Press.
- Mattei, P.-A. and J. Frellsen (2018). “Leveraging the Exact Likelihood of Deep Latent Variable Models”. In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, pp. 3859–3870.

References

- Mattei, P.-A. and J. Frellsen (2019). “MIWAE: Deep Generative Modelling and Imputation of Incomplete Data”. In: *arXiv preprint arXiv:1812.02633*.
- Mohamed, S. and D. Rezende (2017). *Tutorial on Deep Generative Models*. UAI 2017.
- Nazabal, A., P. M. Olmos, Z. Ghahramani, and I. Valera (2018). “Handling incomplete heterogeneous data using VAEs”. In: *arXiv preprint arXiv:1807.03653*.
- Rezende, D. J., S. Mohamed, and D. Wierstra (2014). “Stochastic Backpropagation and Approximate Inference in Deep Generative Models”. In: *Proceedings of the 31st International Conference on Machine Learning*, pp. 1278–1286.