Variational Autoencoders: an introduction to new applications and a new regularization approach.

Cédric Beaulac

Simon Fraser University and University of Victoria

October the 13th 2021

Variational Autoencoders

Formal definition
A survival analysis application
An image analysis application
MEGA: a new moment-matching metric for VAEs

Variational Autoencoders

Formal definition of the model and the training procedure.

What is a Variational Autoencoder ?

- A VAE is a latent variable model like Hidden Markov Models (HMM) and Gaussian Mixture Models (GMM).
- Introduced by Kingma in 2013, it was used with success on image analysis toy examples.
- Not commonly employed by statisticians.
- There have been lots of publications that update and improve the implementation of VAEs.

What is an autoencoder ?

- An AutoEncoder (AE) is an unsupervised learning model that learns how to encode (p) and decode (q) data simultaneously.
- The code is usually of lower dimensions, say $M << D$. Thus, the autoencoder compresses and decompresses high-dimensional data.
- *Notations* : $\mathbf{x}$ are D-dimensional observations, $\mathbf{z}$ is the M-dimensional code, p is the encoding function for $\mathbf{x}$ ($p(\mathbf{x}) = \mathbf{z}$) and q is the decoding function ($q(\mathbf{z}) = \mathbf{x}$).

Autoencoder

- There are multiple possible functions p and q and multiple ways to optimize for those.
- *Specific case*: Assume p and q are linear combinations.
- and assume we minimize the quadratic reconstruction error : $\frac{1}{n}\sum_{i=1}^{n} ||\mathbf{x_i} - \mathbf{\tilde{x}_i}||^2$, where $\mathbf{\tilde{x}} = q(p(x))$.
- Then, the solutions to this problem are the principal components.

Toward a probabilistic autoencoder

- Can we build a probabilistic equivalent ?
- Assume some distributions for both variables:
  1. $p(z) = \mathcal{N}(0, I)$
  2. $p(x|z) = \mathcal{N}(W\mathbf{z} + \mu, \sigma^2 I)$
- This model is called *probabilistic principal component analysis* (pPCA, Tipping & Bishop 1999).
- The marginal distribution of $\mathbf{x}$ is Normal and the parameters W, μ and σ are obtained by maximum likelihood.
- We can analytically compute $p(\mathbf{z}|\mathbf{x})$.

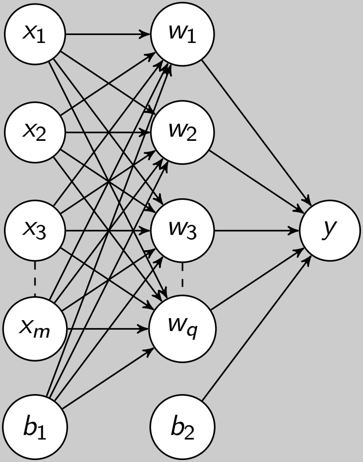
Toward a probabilistic autoencoder

- We now have a *probabilistic encoder* $p(\mathbf{z}|\mathbf{x})$
- and a *probabilistic decoder* $p(\mathbf{x}|\mathbf{z})$.
- The probabilistic formulation offers multiple advantages:
  1. The EM algorithm is fast.
  2. Help manages missing values.
  3. Allows for a Bayesian formulation.
  4. Can model conditional distribution allowing for classification.
  5. Allows generating *new observations* using ancestral sampling.

Toward a variational autoencoder

- VAE is a generalization of pPCA.
- We want to allow for more complex p's and q's.
- A modern flexible function comes in mind: a Neural Network (NN).
- Made of the sequential application of parametric linear combinations and non-linear nonparametric transformations.
- Easy to optimize with back-propagation of the gradient (chain rule of derivatives).
- Is considered to be a *universal function approximator*.

Simple NN: graphical representation



Simple NN: functional representation

$$\mathbf{w} = f\left(\mathbf{B}_1 \mathbf{x}\right) \tag{1}$$

where $\mathbf{B}_1$ is a coefficient matrix and f a non-linear activation function. For instance: $f(a) = \frac{1}{1+e^{-a}}$. Assume the response is a binary variable, then:

$$\tilde{y} = \text{logit}\left(\mathbf{B}_2 f\left(\mathbf{B}_1 \mathbf{x}\right)\right) \tag{2}$$

We can compute the gradients of an error function w.r.t. the parameters ($\mathbf{B}_1$ and $\mathbf{B}_2$) by back-propagation.

Toward a variational autoencoder

- Supposons:
  1. $p_\theta(\mathbf{z}) = \mathcal{N}(0, I)$
  2. $p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mu_x, \sigma_x^2 I)$ where $[\mu_x, \sigma_x] = NN(\mathbf{z})$, say $\theta(\mathbf{z})$.
- The parameters of the emission distribution ($p_\theta(\mathbf{x}|\mathbf{z})$) are the output of NNs taking $\mathbf{z}$ as input.
- Assume θ is the set of parameters of p that requires estimation. $\theta = \{\mu_x(\mathbf{z}), \sigma_x(\mathbf{z})\}$

Toward a variational autoencoder

- This allows us to represent and capture complicated marginal of $\mathbf{x}$ without having to increase the dimension of $\mathbf{z}$.
- Unfortunately, $p_\theta(\mathbf{z}|\mathbf{x})$ is analytically intractable.
- To learn the parameters, we rely on variational Bayes. Assume $q_\varphi(\mathbf{z}|\mathbf{x})$ is a variational approximation of $p_\theta(\mathbf{z}|\mathbf{x})$.
- Assume $q_\varphi(\mathbf{z}|\mathbf{x}) = N(\mu_z, \sigma_z^2 I)$, then $\varphi = \{\mu_z(\mathbf{x}), \sigma_z(\mathbf{x})\}$ is a NN as well. The parameters of the variational distribution ($q_\varphi(\mathbf{z}|\mathbf{x})$) are the output of NNs taking $\mathbf{x}$ as input.

ELBO

- It is impossible to directly maximize $\log p_\theta(\mathbf{x})$ or to use EM ($p_\theta(\mathbf{z}|\mathbf{x})$ being intractable).
- Thus the common solution is to optimize a lower bound of $\log p_\theta(\mathbf{x})$, the ELBO (*Evidence Lower BOund*).

ELBO

$$
\begin{aligned}
\log p(x) &= \mathbf{E}_{q(z|x)}[\log p(x)] \\
&= \mathbf{E}_{q_\varphi(z|x)}\left[\log\left(\frac{p(x,z)}{p(z|x)}\right)\right] \\
&= \mathbf{E}_{q(z|x)}\left[\log\left(\frac{p(x,z)q(z|x)}{q(z|x)p(z|x)}\right)\right] \\
&= \mathbf{E}_{q(z|x)}\left[\log\left(\frac{p(x,z)}{q(z|x)}\right)\right] - \mathbf{E}_{q(z|x)}\left[\log\left(\frac{p(z|x)}{q(z|x)}\right)\right] \\
&= \mathcal{L}(q_\varphi, p_\theta) + KL(q_\varphi || p_\theta).
\end{aligned} \tag{3}
$$

ELBO

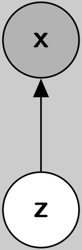
$$\mathcal{L}(q_\varphi, p_\theta) = \mathbf{E}_{q_\varphi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{z}) + \log p_\theta(\mathbf{x}|\mathbf{z}) - \log q_\varphi(\mathbf{z}|\mathbf{x})\right] \tag{4}$$

- The gap between $\log p(\mathbf{x})$ and $\mathcal{L}(q_\varphi, p_\theta)$ is $KL(q_\varphi || p_\theta)$
- Since it is impossible to analytically compute the expectation we estimate it by Monte Carlo.

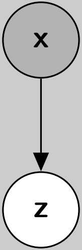
VAE : Algorithm

Algorithm : Training VAE($\mathbf{x}$)
1) Input $\mathbf{x}$ into the NN φ to retrieve $\mu_z(\mathbf{x})$ and $\sigma_z(\mathbf{x})$ 2) Sample $\mathbf{z}$ from $q_{\varphi(x)}(\mathbf{z}\|\mathbf{x})$ 3) Input the sample $\mathbf{z}$ in the NN θ to retrieve $\mu_x(\mathbf{z})$ and $\sigma_x(\mathbf{z})$ 4) Evaluate $\log p_\theta(\mathbf{z}) + \log p_\theta(\mathbf{x}\|\mathbf{z}) - \log q_\varphi(\mathbf{z}\|\mathbf{x})$ 5) Maximise the ELBO Monte Carlo estimate w.r.t the parameters of φ and θ using any gradient-based algorithm Repeat 1-5 until convergence.

VAE: graphical representation



(a) Generative model $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|z)$.



(b) Inference model. Given $\mathbf{x}$ we have $q(\mathbf{z}|\mathbf{x})$.

Figure: Graphical representation of both components of a VAE

VAE: practical uses

- Compression, encoding, storage and latent space analysis.
- Generation of new observation using ancestral sampling: $z \sim p_\theta(\mathbf{z})$ then $x \sim p_\theta(\mathbf{x}|\mathbf{z})$.
- Classification and regression. The model can be adapted for supervised tasks.

Variational autoencoder

Survival analysis application

Introduction

- We received a data set from the *Children's Oncology Group*.
- It consists of 1 712 patients. We have patient symptoms as well as the treatment and the response.
- The response is a time-to-event variable that is right-censored for the majority of patients.
- We want a system that recommends treatment based on patient symptoms.
- Work published in the *2018 NeurIPS ML4H workshop* et in *Applied Artificial Intelligence*.

Our model: SAVAE (Survival Analysis VAE)

(a) Generative model. Assume $p(x, y, t, z) = p(z)p(x|z)p(t|x)p(y|t, z)$.

(b) Inference model. Given x and y we have $q(z|x, y)$.

Figure: Graphical representation where y is the response, t is the treatment, $\mathbf{x}$ are the characteristics and symptoms and $\mathbf{z}$ is the latent variable which represents the true health status of the patient.

SAVAE

$$\begin{aligned}\text{ELBO} &= \mathbf{E}_{q_\varphi}\left[\log \frac{p_\theta(\mathbf{x}, t, y, \mathbf{z})}{q_\varphi(\mathbf{z}|\mathbf{x}, y)}\right] = \mathbf{E}_{q_\varphi}\left[\log p_\theta(\mathbf{x}, t, y, \mathbf{z}) - \log q_\varphi(\mathbf{z}|\mathbf{x}, y)\right] \\ &= \mathbf{E}_{q_\varphi}[\log p_\theta(\mathbf{z}) + \log p_\theta(\mathbf{x}|\mathbf{z}) + \log p_\theta(t|\mathbf{x}) + \log p_\theta(y|t, \mathbf{z}) \\ &- \log q_\varphi(\mathbf{z}|\mathbf{x}, y)].\end{aligned} \tag{5}$$

where

$$\log p_\theta(y|t, \mathbf{z}) = \delta \log f_\theta(y|t, \mathbf{z}) + (1 - \delta) \log S_\theta(y|t, \mathbf{z}), \tag{6}$$

with $\delta = 1$ if y is observed et 0 if y is censored.

SAVAE

We select the distributions.

$$p_\theta(\mathbf{x}|\mathbf{z}) = \prod_{j=1}^{D_x} p_\theta(x_j|z) \tag{7}$$

$$p(t_i|\mathbf{x}) = \text{Ber}(\hat{\pi}_i) \text{ pour } i \in \{1, 2\}. \tag{8}$$

$$p(y|t, \mathbf{z}) = \text{Weibull}(\lambda, K) \tag{9}$$

$$\theta = f_2(\mathbf{B}_2 f_1(\mathbf{B}_1 \mathbf{z})) \tag{10}$$

$$[\pi_1, \pi_2] = f_4(\mathbf{B}_4 f_3(\mathbf{B}_3 \mathbf{x})) \tag{11}$$

$$[\lambda, K] = f_6(\mathbf{B}_6 f_5(\mathbf{B}_5 [t, \mathbf{z}])) \tag{12}$$

SAVAE

$$q(\mathbf{z}|\mathbf{x}, y) = \mathcal{N}(\mu, \sigma^2 I) \tag{13}$$

$$[\mu, \sigma] = f_8(\mathbf{B}_8 f_7(\mathbf{B}_7 [x, y])). \tag{14}$$

SAVAE

Finally, we obtain $p(y|t, \mathbf{x})$ by importance sampling:

$$p(y|t, x) \approx \sum_{l=1}^{L} w_l p_\theta(y|t, \mathbf{z}_l) \tag{15}$$

where:

$$w_l = \frac{p_\theta(\mathbf{x}|\mathbf{z}_l)}{\sum_{k=1}^{L} p_\theta(\mathbf{x}|\mathbf{z}_k)} \tag{16}$$

Results

- Performed better than Cox regression according to the Brier score.
- Provides a completely defined a Weibull survival distribution for every possible patient and treatment combination.
- This allows the physician to select the treatment in different ways.

Variational autoencoder

Image analysis application

Introduction

- I love imagine analysis and I wanted to explore the topic during my Ph.D.
- Contributions: a new database and a related analysis
- Paper under review at the moment with *Springer Nature: Compute Science*.

Motivation

Inspired by the popular *MNIST data set*.

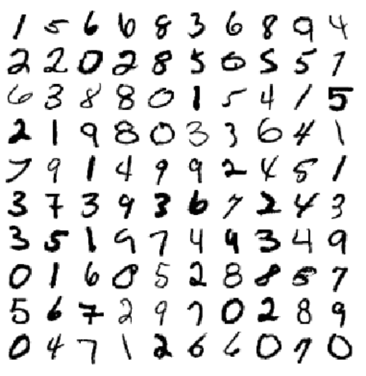


Figure: Samples of images from the *MNIST data set*.

Motivation

When fitting a VAE on those images (with a 2-dimensional latent space), we see that digits with similar styles are clustered together.

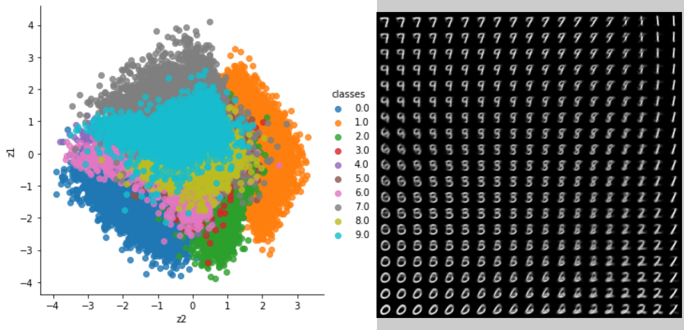


Figure: Latent representation of the *MNIST data set*.

Motivation

- Since writing styles depend on the writer, can we predict writers ?
- MNIST is a *too simple* data set:
  1. Contains images of low resolution.
  2. Contains only the digit as response.
  3. Easy to achieve high accuracy.
- Thus, we decided to collect our own data set:
  1. Can we determine the digit writers ?
  2. Can we predict writer characteristics such as age and gender ?
  3. Finally, can we generate new images where we control the digit and its style ?

Data gathering

- Our goal: 200-300 students at UofT
- We booked multiple classrooms over a few days.
- For March 2020...
- Settled on mail instead, ended up with 97 participants.

Data gathering

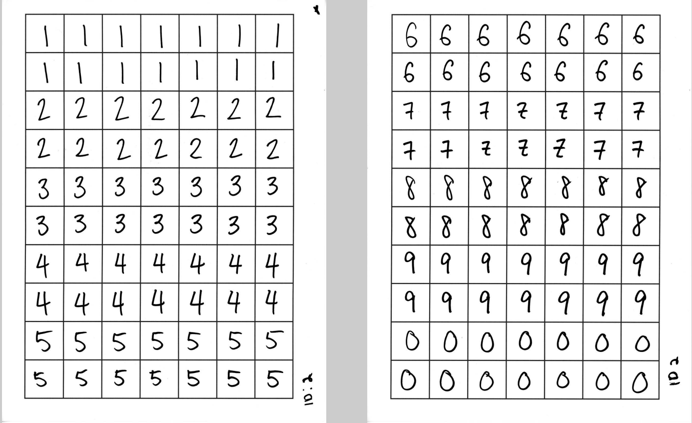


Figure: Examples of the collected data sheets.

Data: general information

- 97 writers, 14 occurrences for all 10 digits for a total of 13 580 images in high resolution (500 x 500).
- We gathered: the digit, writer ID, age, biological gender, height,native language, handiness, education level and main writing medium.
- Publicly available on my website.
- Available in multiple formats.

Data: a sample

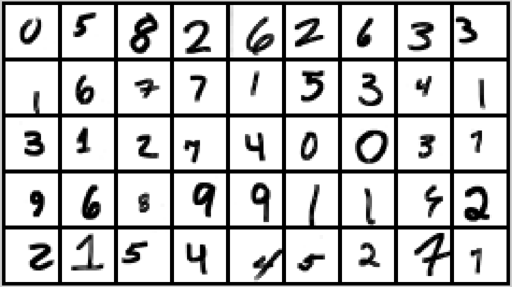


Figure: Samples of 45 images.

Questions specific to our data set

1. Can we predict the digit (easy task), the ID (much more difficult) or other characteristics ?
2. What is the impact of the image resolution?
3. How does semi-supervised prediction works ?
4. Can we do controlled image generation ?

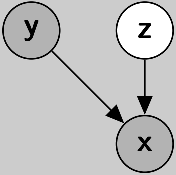
Results

- For (1) and (2) our new data set provides new opportunity compared to what MNIST offers.
- But let's focus on the VAE applications: (3) and (4).

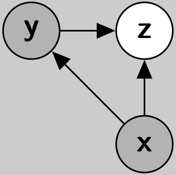
Semi-supervised learning

- Can we incorporate unlabelled data (S_u) to a data set with labelled observations (S_l) to improve the prediction accuracy.
- Our data set is different of *MNIST*, but similar enough for these experiments.
- We can check if our predictions are more accurate when integrating unlabelled MNIST data.
- We use the VAE M2 (Kingma 2014)

VAE: M2



(a) Generative component. Assumes $p_\theta(\mathbf{x}, \mathbf{z}, \mathbf{y}) = p_\theta(\mathbf{z}) p_\theta(\mathbf{y}) p_\theta(\mathbf{x}|z, y)$.



(b) Inference network. Given x and y we get $q_\varphi(\mathbf{z}|x, y)$. If y is missing, we can estimate it with $q_\varphi(\mathbf{y}|x)$.

Figure: Graphical representation of M2.

VAE: M2

$$\begin{aligned}\log p_\theta(\mathbf{x},\mathbf{y}) &\geq \mathbf{E}_{q(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{z}) + p_\theta(\mathbf{y}) + p_\theta(\mathbf{x}|z,y) - \log q_\varphi(\mathbf{z}|x,y)\right] \\ &= \mathcal{L}(x,y)\end{aligned} \tag{17}$$

$$\begin{aligned}\log p_\theta(\mathbf{x}) &\geq \mathbf{E}_{q(\mathbf{z},\mathbf{y}|x)}\left[\log p_\theta(\mathbf{z}) + p_\theta(\mathbf{y}) + p_\theta(\mathbf{x}|z,y) - \log q_\varphi(\mathbf{z},\mathbf{y}|x)\right] \\ &= \sum_y \left[q_\varphi(\mathbf{y}|x)(\mathcal{L}(x,y))\right] + \mathcal{H}(q_\varphi(\mathbf{y}|x)) \\ &= \mathcal{U}(x)\end{aligned} \tag{18}$$

$$\mathcal{J} = \sum_{S_l} \mathcal{L}(x,y) + \sum_{S_u} \mathcal{U}(x) \tag{19}$$

VAE: M2

However, what is strange about using the ELBO here is that we train $q_\varphi(\mathbf{y}|x)$ (here a CNN) only using unlabelled data. The solution proposed (Kingma 2014) is to modify the objective function :

$$\mathcal{J}^\alpha = \mathcal{J} + \alpha \mathbf{E}_{S_l}\left[\log q_\varphi(\mathbf{y}|x)\right] \tag{20}$$

Notes: These heuristic modifications are made over and over again in the ML literature, I'd like to establish more formal definitions for these.

Possible explanation: to be explored

$$\mathcal{J}^{\alpha} = \alpha\mathcal{J} + \mathbf{E}_{S_l}\left[\log q_{\varphi}(\mathbf{y}|x)\right] \tag{21}$$

When $\alpha = 0$ we basically train a supervised model.

It seems like the unsupervised VAE *machinerie* act as regularizer.

Semi-supervised classification: results

	CNN		M2	
	Mean	Std.	Mean	Std.
Digit	0.9399	0.0143	**0.9542**	0.0060
ID	0.3473	0.0136	**0.4174**	0.0099

Table: Prediction accuracy for two classification problems.

Image generation

- A VAE is a generative model. Since $p(x, z)$ is fully defined and estimated, we can sample from it and generate new observations x.
- In this case, it means generating new images.
- With a simple VAE it means $z \sim p(z)$ then $x \sim p(x|z)$.
- This process generates images of a random digit and random style.

Controlled generation

- Can we decide on the digit or the style ? Yes, using a VAE designed for classification, such as M2 defined earlier.
- In this case: we fix y then $z \sim p(z)$ and $x \sim p(x|z, y)$.
- *Our Assumption:* If there exist a signal between the variable and the image, then we can use it to control the content of the image.

Controlled generation: Results

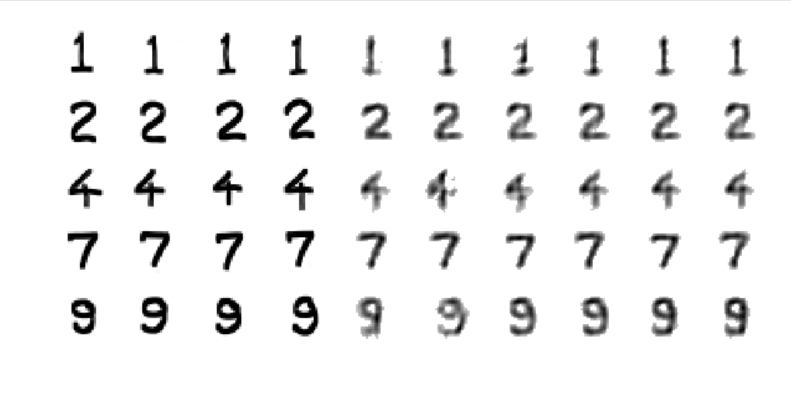


Figure: Examples of controlled image generation.

Controlled generation: Results

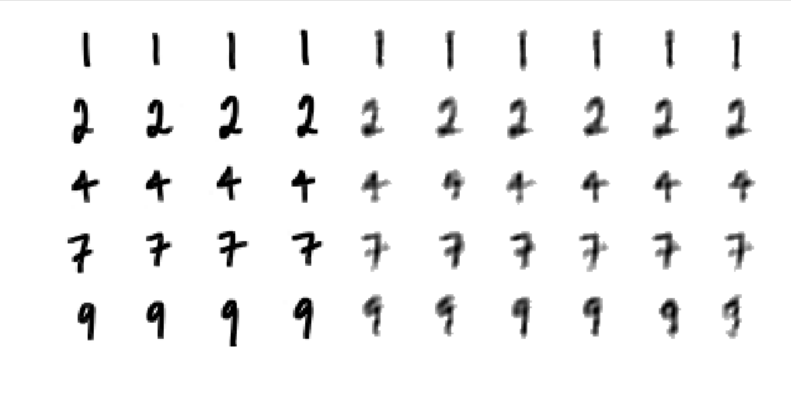


Figure: Examples of controlled image generation.

Controlled generation: Results

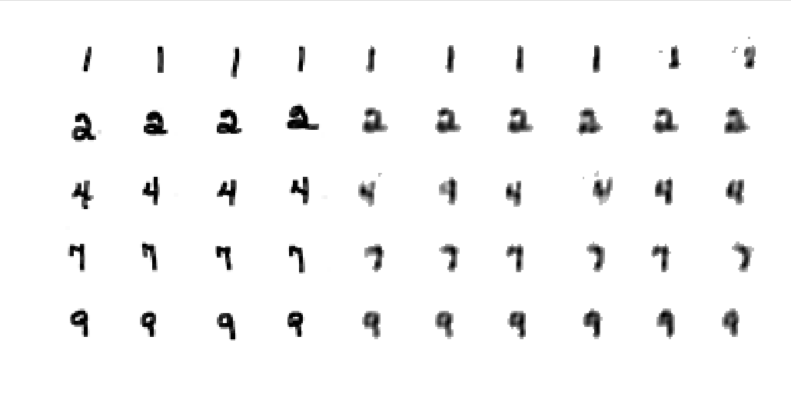


Figure: Examples of controlled image generation.

Controlled generation: Results

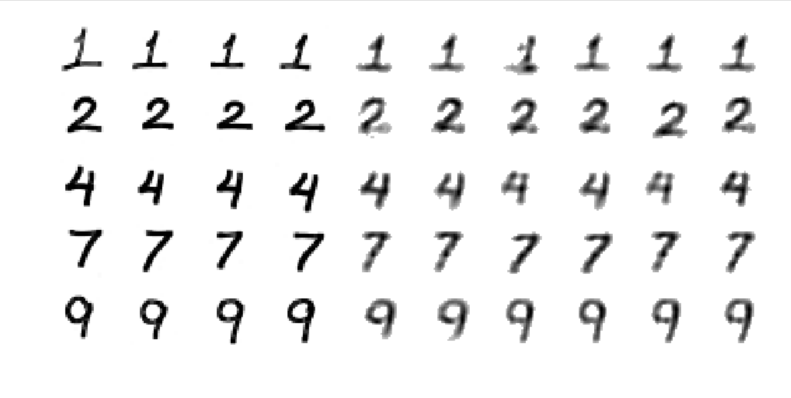
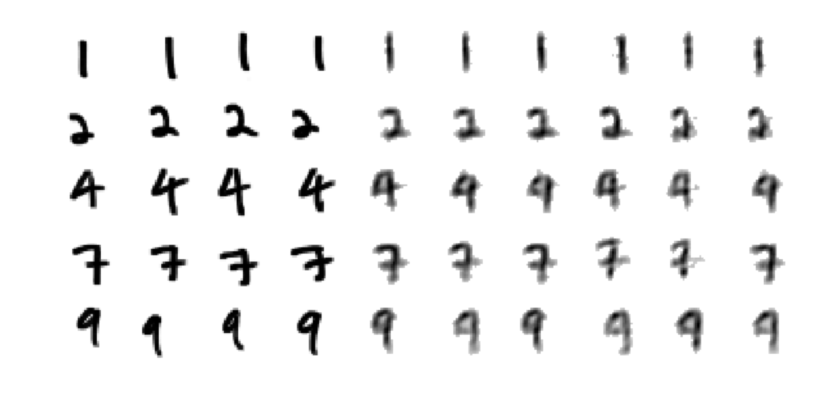


Figure: Examples of controlled image generation.

Controlled generation: Results



Moment Estimators GAp (MEGA)

A new metric for comparing or regularizing models

Moment Estimators GAp (MEGA)

- A new metric to compare or regularized latent variable generative models.
- Project I have been working on after I noticed some issues with implementing the theoretical VAE.
- Paper just submitted to *Journal of Machine Learning Research*

Assessing an unsupervised model

- In supervised learning we learn $p(y|x)$ and we can check our results against unobserved data points (x, y).
- In supervised learning, there are no labels y and we simply try to fit $p(x)$. It is much more complicated to assess the quality of the fit.
- Parametric models are fitted by maximum likelihood so we cannot use the likelihood to compare these models.

Assessing an unsupervised model

- We propose a new metric based on moments that is suitable to compare any latent variable generative models, such as GMMs and VAEs.
- It is fast to compute and provides a good sanity check.
- We also demonstrate how to use such metric to regularize such models. However, it can no longer be used for model comparison.

MEGA: Key concept

- We compare two estimators of the second moment of $p(\mathbf{x})$; one comes from the data, the other from the trained model.
- Using the Law of Total Variance:

$$\mathbf{Var}_x(\mathbf{x}) = \mathbf{E}_z[\mathbf{Var}_x(\mathbf{x}|\mathbf{z})] + \mathbf{Var}_z[\mathbf{E}_x(\mathbf{x}|\mathbf{z})], \tag{22}$$

and notice the second term is

$$\begin{aligned} \mathbf{Var}_z[\mathbf{E}_x(\mathbf{x}|\mathbf{z})] &= \mathbf{E}_z[\mathbf{E}_x(\mathbf{x}|\mathbf{z})^2] - (\mathbf{E}_z[\mathbf{E}_x(\mathbf{x}|\mathbf{z})])^2 \qquad (23) \\ &= \mathbf{E}_z[\mathbf{E}_x(\mathbf{x}|\mathbf{z})^2] - (\mathbf{E}_x[\mathbf{x}])^2. \qquad (24) \end{aligned}$$

We combine and reorganize both equations

$$\mathbf{Var}_x(\mathbf{x}) + (\mathbf{E}_x[\mathbf{x}])^2 = \mathbf{E}_z[\mathbf{Var}_x(\mathbf{x}|\mathbf{z})] + \mathbf{E}_z[\mathbf{E}_x(\mathbf{x}|\mathbf{z})^2]. \tag{25}$$

Both sides are the equal to the second moment of $\mathbf{x}$.

MEGA: Moment estimators

Data estimator:

$$\mathbf{Var}_x(\mathbf{x}) + (\mathbf{E}_x[\mathbf{x}])^2 \approx \frac{\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})^T(\mathbf{x}_i - \bar{x})}{n-1} + \bar{\mathbf{x}}^T\bar{\mathbf{x}} := \text{DE} \quad (26)$$

Forward model estimator:

$$\begin{aligned}\mathbf{E}_z[\mathbf{Var}_x(\mathbf{x}|\mathbf{z}) + \mathbf{E}_x(\mathbf{x}|\mathbf{z})^2] &= \int_z \left[\mathbf{Var}_x(\mathbf{x}|\mathbf{z}) + \mathbf{E}_x(\mathbf{x}|\mathbf{z})^2\right] p(z)dz \\ &\approx \frac{1}{m}\sum_{i=1}^{m}[\mathbf{Var}_x(\mathbf{x}|\mathbf{z} = z_i) \\ &+ \mathbf{E}_x(\mathbf{x}|\mathbf{z} = z_i)^T\mathbf{E}_x(\mathbf{x}|\mathbf{z} = z_i)] := \text{FME}\end{aligned} \quad (27)$$

MEGA: Compute the gap

- The gap between those two moment estimators is DE-FME.
- The bigger this gap is the further the model is from the observed second moment.
- Those are 2-dimensional matrices.
- We are using matrix norms to make the gap more digestible.

MEGA: Frobenius norm

- Schatten q-norm is a well-studied family of matrix norms with:

$$|M|_q = (\sum_{ij} |M_{ij}|^q)^{(1/q)}. \tag{28}$$

- When $q = 2$, this is a special case called the Frobenius norm:

$$|M|_2 = |M|_F = (\sum_{ij} |M_{ij}|^2)^{(1/2)} = \sqrt{\mathrm{Tr}(M^T M)}. \tag{29}$$

- Thus the proposed metric is:

$$2\text{MEGA-F} = |\text{DE-FME}|_F. \tag{30}$$

MEGA for regularization

- Because our metric favours simple model, such as a single Gaussian. It can be used for as a regularizer.
- For GMMs, it behaves similarly to the AIC or the BIC.
- We can also use it to regularize VAEs

MEGA for VAE regularization

VAE

$$\mathcal{L}(q_\varphi, p_\theta) = \mathbf{E}_{q_\varphi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{z}) + \log p_\theta(\mathbf{x}|\mathbf{z}) - \log q_\varphi(\mathbf{z}|\mathbf{x})\right] \quad (31)$$

$$= \mathbf{E}_{q_\varphi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - KL(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z})) \quad (32)$$

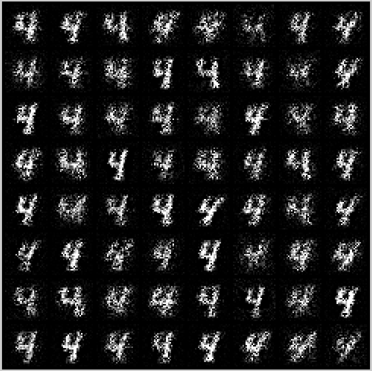
β-VAE

$$\underbrace{\mathbf{E}_q[\ln p(\mathbf{x}|\mathbf{z})]}_{\text{Reconstruction error}} - \underbrace{\beta KL(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))}_{\text{Regularization for} q(\mathbf{z}|\mathbf{x})} \quad (33)$$

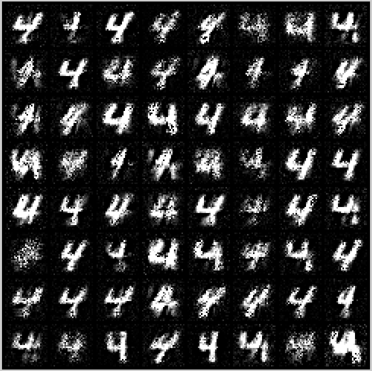
MEGA-β-VAE

$$\underbrace{\mathbf{E}_q[\ln p(\mathbf{x}|\mathbf{z})]}_{\text{Reconstruction error}} - \underbrace{\beta KL(q(\mathbf{z}|\mathbf{x})|p(\mathbf{z}))}_{\text{Regularization for} q(\mathbf{z}|\mathbf{x})} - \underbrace{\alpha(2\text{MEGA-F})}_{\text{Regularization for} p(\mathbf{x})} \quad (34)$$

MEGA for regularization: results



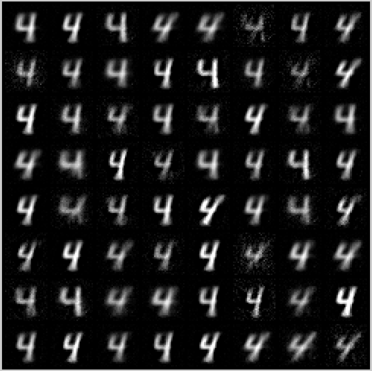
(a) Model train without MEGA



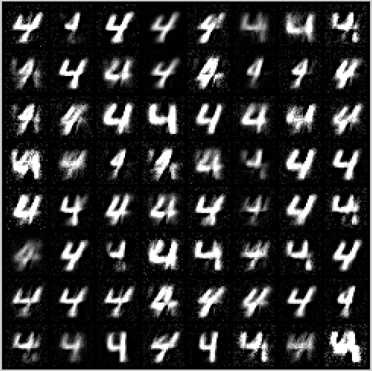
(b) Model train with MEGA

Figure: A sample of 64 images from $p_\theta(\mathbf{x}\|\mathbf{z}) = N(\mu(\mathbf{z}), \sigma(\mathbf{z}))$ where $\mathbf{z} \sim N(0, 1)$.

MEGA for regularization: results



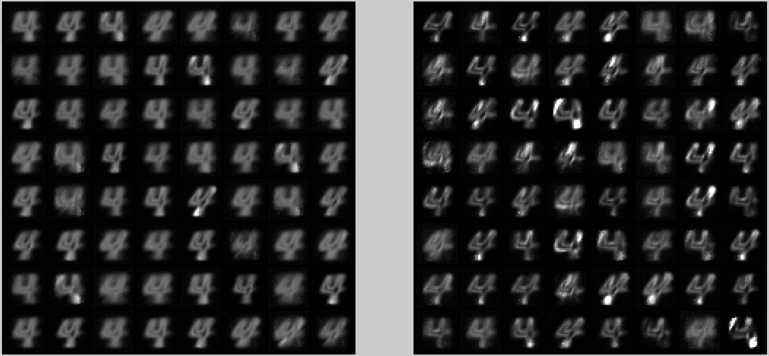
(a) Model train without MEGA



(b) Model train with MEGA

Figure: The 64 sampled means of the images: $\mu(\mathbf{z})$ where $\mathbf{z} \sim N(0, 1)$.

MEGA for regularization: results



(a) Model train without MEGA

(b) Model train with MEGA

Figure: The 64 sampled standard deviation for each pixel of the images: $\sigma(\mathbf{z})$ where $\mathbf{z} \sim N(0, 1)$. For those images, the whiter the pixel is the larger the standard deviation of that pixel is.

I would love to answer your questions.

Beaulac, C., Rosenthal, J. S., & Hodgson, D. (2018). A deep latent-variable model application to select treatment intensity in survival analysis. MI4H Workshop, NeurIPS 2018.

Beaulac, C., Rosenthal, J. S., Pei, Q., Friedman, D., Wolden, S., & Hodgson, D. (2020). An evaluation of machine learning techniques to predict the outcome of children treated for Hodgkin-Lymphoma on the AHOD0031 trial. Applied Artificial Intelligence, 1-15.

Beaulac, C. & Rosenthal (2020). Analysis of a high-resolution hand-written digits data set with writer characteristics, pre-print.

Beaulac, C. (2021). A new moment-matching metric for latent variable generative models.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. Proceedings of the 2nd International Conference on Learning Representations (ICLR)

Kingma, D. P., Mohamed, S., Rezende, D. J., & Welling, M. (2014). Semi-supervised learning with deep generative models. In Advances in neural information processing systems (pp. 3581-3589).

Tipping, M. E., & Bishop, C. M. (1999). Probabilistic principal component analysis. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 61(3), 611-622.