Deep Generative Models
Lecture 9

Roman Isachenko

Ozon Masters

Spring, 2021

Recap of previous lecture

ELBO surgery

$$\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_i(q,\boldsymbol{\theta}) = \underbrace{\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)}\log p(\mathbf{x}_i|\mathbf{z},\boldsymbol{\theta})}_{\text{Reconstruction loss}} - \underbrace{\mathbb{I}_q[\mathbf{x},\mathbf{z}]}_{\text{MI}} - \underbrace{KL(q(\mathbf{z})||p(\mathbf{z}))}_{\text{Marginal KL}}$$

Optimal prior

$$KL(q(\mathbf{z})||p(\mathbf{z})) = 0 \quad \Leftrightarrow \quad p(\mathbf{z}) = q(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n} q(\mathbf{z}|\mathbf{x}_i).$$

The optimal prior distribution $p(\mathbf{z})$ is aggregated posterior $q(\mathbf{z})$.

VampPrior

$$p(\mathbf{z}|\boldsymbol{\lambda}) = \frac{1}{K}\sum_{k=1}^{K} q(\mathbf{z}|\mathbf{u}_k),$$

where $\boldsymbol{\lambda} = \{\mathbf{u}_1, \dots, \mathbf{u}_K\}$ are trainable pseudo-inputs.

Tomczak J. M., Welling M. VAE with a VampPrior, 2017

Recap of previous lecture

Autoregressive flow prior

$$\log p(\mathbf{z}|\boldsymbol{\lambda}) = \log p(\boldsymbol{\epsilon}) + \log \det \left| \frac{d\boldsymbol{\epsilon}}{d\mathbf{z}} \right|; \quad \mathbf{z} = g(\boldsymbol{\epsilon}, \boldsymbol{\lambda}) = f^{-1}(\boldsymbol{\epsilon}, \boldsymbol{\lambda})$$

Theorem

VAE with the AF prior for latent code $\mathbf{z}$ is equivalent to using the IAF posterior for latent code $\boldsymbol{\epsilon}$.

$$\mathcal{L}(q, \boldsymbol{\theta}) = \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} \Bigg[\log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) + \underbrace{\left(\log p(f(\mathbf{z}, \boldsymbol{\lambda})) + \log \left| \det \frac{\partial f(\mathbf{z}, \boldsymbol{\lambda})}{\partial \mathbf{z}} \right| \right)}_{\text{AF prior}} - \log q(\mathbf{z}|\mathbf{x}) \Bigg]$$

$$= \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} \Bigg[\log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) + \log p(f(\mathbf{z}, \boldsymbol{\lambda})) - \underbrace{\left(\log q(\mathbf{z}|\mathbf{x}) - \log \left| \det \frac{\partial f(\mathbf{z}, \boldsymbol{\lambda})}{\partial \mathbf{z}} \right| \right)}_{\text{IAF posterior}} \Bigg]$$

Chen X. et al. Variational Lossy Autoencoder, 2016

Recap of previous lecture

LVM

$$p(\mathbf{x}|\boldsymbol{\theta}) = \int p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) d\mathbf{z} = \int p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) p(\mathbf{z}) d\mathbf{z}$$

- More powerful $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$ leads to more powerful generative model $p(\mathbf{x}|\boldsymbol{\theta})$.
- Too powerful $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$ could lead to posterior collapse: $q(\mathbf{z}|\mathbf{x})$ will not carry any information about $\mathbf{x}$ and close to prior $p(\mathbf{z})$.

Autoregressive decoder

$$p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \prod_{i=1}^{n} p(x_i|\mathbf{x}_{1:i-1}, \mathbf{z}, \boldsymbol{\theta})$$

- Global structure is captured by latent variables.
- Local statistics are captured by limited receptive field autoregressive model.

Gulrajani I. et al. PixelVAE: A Latent Variable Model for Natural Images, 2016

Recap of previous lecture

Decoder weakening

- Powerful decoder $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$ makes the model expressive, but posterior collapse is possible.
- PixelVAE model uses the autoregressive PixelCNN model with small number of layers to limit receptive field.

KL annealing

$$\mathcal{L}(q, \boldsymbol{\theta}, \beta) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - \beta \cdot KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$$

Start training with $\beta = 0$, increase it until $\beta = 1$ during training.

Free bits

Ensure the use of less than λ bits of information:

$$\mathcal{L}(q, \boldsymbol{\theta}, \lambda) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - \max(\lambda, KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))).$$

This results in $KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \geq \lambda$.

Recap of previous lecture

Disentanglement learning

A disentangled representation is a one where single latent units are sensitive to changes in single generative factors, while being invariant to changes in other factors.

β-VAE

$$\mathcal{L}(q, \boldsymbol{\theta}, \beta) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - \beta \cdot KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})).$$

Representations becomes disentangled by setting a stronger constraint with $\beta > 1$. However, it leads to poorer reconstructions and a loss of high frequency details.

ELBO surgery

$$\frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_i(q, \boldsymbol{\theta}, \beta) = \underbrace{\frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)} \log p(\mathbf{x}_i|\mathbf{z}, \boldsymbol{\theta})}_{\text{Reconstruction loss}} - \beta \cdot \underbrace{\mathbb{I}_q[\mathbf{x}, \mathbf{z}]}_{\text{MI}} - \beta \cdot \underbrace{KL(q(\mathbf{z})||p(\mathbf{z}))}_{\text{Marginal KL}}$$

Burgess C. P. et al. Understanding disentangling in β-VAE, 2018

DIP-VAE

Disentangled aggregated variational posterior

$$q(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n} q(\mathbf{z}|\mathbf{x}) = \prod_{j=1}^{d} q(z_j)$$

DIP-VAE Objective

$$\mathcal{L}_{\text{DIP}}(q, \boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_i(q, \boldsymbol{\theta}) - \lambda \cdot KL(q(\mathbf{z})||p(\mathbf{z})) =$$

$$= \frac{1}{n}\sum_{i=1}^{n} \left[\mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)} \log p(\mathbf{x}_i|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z}|\mathbf{x}_i)||p(\mathbf{z}))\right] - \lambda \cdot KL(q(\mathbf{z})||p(\mathbf{z})) =$$

$$= \underbrace{\frac{1}{n}\sum_{i=1}^{n} \left[\mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)} \log p(\mathbf{x}_i|\mathbf{z}, \boldsymbol{\theta})\right]}_{\text{Reconstruction loss}} - \underbrace{\mathbb{I}_q[\mathbf{x}, \mathbf{z}]}_{\text{MI}} - (1 + \lambda) \cdot \underbrace{KL(q(\mathbf{z})||p(\mathbf{z}))}_{\text{Marginal KL}}$$

Kumar A., Sattigeri P., Balakrishnan A. Variational Inference of Disentangled Latent Concepts from Unlabeled Observations, 2017

DIP-VAE

$$\mathcal{L}_{\text{DIP}}(q, \boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_i(q, \boldsymbol{\theta}) - \lambda \cdot \underbrace{KL(q(\mathbf{z}) || p(\mathbf{z}))}_{\text{intractable}}$$

Let match the moments of $q(\mathbf{z})$ and $p(\mathbf{z})$:

$$\text{cov}_{q(\mathbf{z})}(\mathbf{z}) = \mathbb{E}_{q(\mathbf{z})} \left[(\mathbf{z} - \mathbb{E}_{q(\mathbf{z})}(\mathbf{z}))(\mathbf{z} - \mathbb{E}_{q(\mathbf{z})}(\mathbf{z}))^T \right]$$

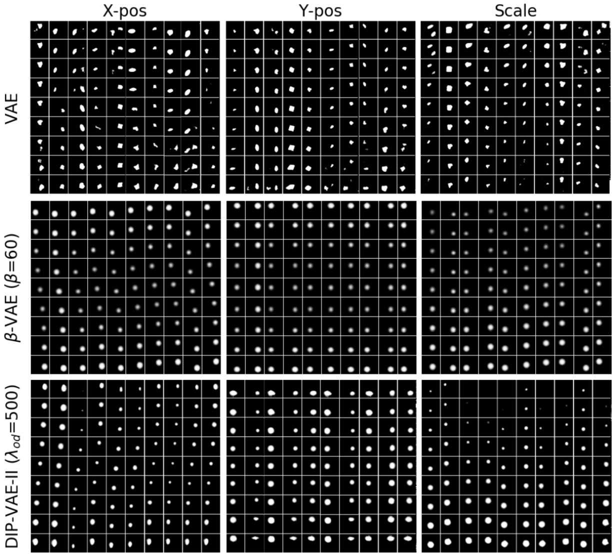
DIP-VAE regularizes $\text{cov}_{q(\mathbf{z})}(\mathbf{z})$ to be close to the identity matrix.

Objective

$$\max_{q, \boldsymbol{\theta}} \Big[\frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_i(q, \boldsymbol{\theta}) - \\ - \lambda_1 \sum_{i \neq j} \left[\text{cov}_{q(\mathbf{z})}(\mathbf{z}) \right]_{ij}^2 - \lambda_2 \sum_{i} \left(\left[\text{cov}_{q(\mathbf{z})}(\mathbf{z}) \right]_{ii} - 1 \right)^2 \Big]$$

Kumar A., Sattigeri P., Balakrishnan A. Variational Inference of Disentangled Latent Concepts from Unlabeled Observations, 2017

DIP-VAE



Kumar A., Sattigeri P., Balakrishnan A. Variational Inference of Disentangled Latent Concepts from Unlabeled Observations, 2017

Challenging Disentanglement Assumptions

Whether unsupervised disentanglement learning is even possible for arbitrary generative models?

Theorem

For $d > 1$, let $\mathbf{z} \sim P$ denote any distribution which admits a density $p(\mathbf{z}) = \prod_{i=1}^{d} p(z_i)$. Then, there exists an infinite family of bijective functions $f : \mathrm{supp}(\mathbf{z}) \to \mathrm{supp}(\mathbf{z})$ such that

- $\frac{\partial f_i(\mathbf{z})}{\partial z_j} \neq 0$ almost everywhere for all i and j (i.e., $\mathbf{z}$ and $f(\mathbf{z})$ are completely entangled);
- and $P(\mathbf{z} \leq \mathbf{u}) = P(f(\mathbf{z}) \leq \mathbf{u})$ for all $\mathbf{u} \in \mathrm{supp}(\mathbf{z})$ (i.e., they have the same marginal distribution).

Theorem claims that unsupervised disentanglement learning is impossible for arbitrary generative models with a factorized prior.

Locatello F. et al. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations, 2018

Challenging Disentanglement Assumptions

Assume we have $p(\mathbf{z})$ and some $p(\mathbf{x}|\mathbf{z})$ defining a generative model. Consider any unsupervised disentanglement method and assume that it finds a representation that is perfectly disentangled with respect to $\mathbf{z}$ in the generative model.

- Theorem claims that $\exists\ \hat{\mathbf{z}} = f(\mathbf{z})$ where $\hat{\mathbf{z}}$ is completely entangled with respect to $\mathbf{z}$.
- Since the (unsupervised) disentanglement method only has access to observations $\mathbf{x}$, it hence cannot distinguish between the two equivalent generative models and thus has to be entangled to at least one of them

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} = \int p(\mathbf{x}|\hat{\mathbf{z}})p(\hat{\mathbf{z}})d\hat{\mathbf{z}}.$$

Locatello F. et al. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations, 2018

Challenging Disentanglement Assumptions

Proof (1)

1. Consider the function $g : \text{supp}(\mathbf{z}) \to [0,1]^d$:

$$g_i(\mathbf{u}) = P(z_i \leq u_i), \quad i = 1, \dots, d.$$

   - g is bijective (since $p(\mathbf{z}) = \prod_{i=1}^{d} p(z_i)$).
   - $\frac{\partial g_i(\mathbf{u})}{\partial u_i} \neq 0$, for all i and $\frac{\partial g_i(\mathbf{u})}{\partial u_j} = 0$ for all $i \neq j$.
   - $g(\mathbf{z})$ is an independent d-dimensional uniform distribution.

2. Consider $h : (0,1]^d \to \mathbb{R}^d$

$$h_i(\mathbf{u}) = \psi^{-1}(u_i), \quad i = 1, \dots, d.$$

   Here ψ denotes the CDF of a standard normal distribution.

   - h is bijective.
   - $\frac{\partial h_i(\mathbf{u})}{\partial u_i} \neq 0$, for all i and $\frac{\partial h_i(\mathbf{u})}{\partial u_j} = 0$ for all $i \neq j$.
   - $h(g(\mathbf{z}))$ is a d-dimensional standard normal distribution.

Locatello F. et al. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations, 2018

Challenging Disentanglement Assumptions

Proof (2)

Let $\mathbf{A} \in \mathbb{R}^{d \times d}$ be an arbitrary orthogonal matrix with $A_{ij} \neq 0$ for all i, j. The family of such matrices is infinite.

- $\mathbf{A}$ is orthogonal, it is invertible and thus defines a bijective linear operator.
- $\mathbf{A}h(g(\mathbf{z})) \in \mathbb{R}^d$ is hence an independent, multivariate standard normal distribution.
- $h^{-1}(\mathbf{A}h(g(\mathbf{z}))) \in \mathbb{R}^d$ is an independent d-dimensional uniform distribution.

Define f : supp($\mathbf{z}$) $\rightarrow$ supp($\mathbf{z}$):

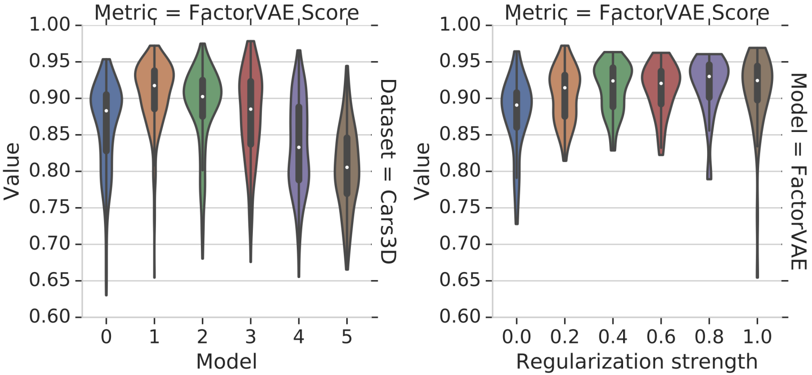
$$f(\mathbf{u}) = g^{-1}(h^{-1}(\mathbf{A}h(g(\mathbf{z})))).$$

By definition $f(\mathbf{z})$ has the same marginal distribution as $\mathbf{z}$: $P(\mathbf{z} \leq \mathbf{u}) = P(f(\mathbf{z}) \leq \mathbf{u})$ and $\frac{\partial f_i(\mathbf{z})}{\partial z_j} \neq 0$.

Locatello F. et al. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations, 2018

Challenging Disentanglement Assumptions

Importance of different models and hyperparameters for disentanglement



Locatello F. et al. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations, 2018

Challenging Disentanglement Assumptions

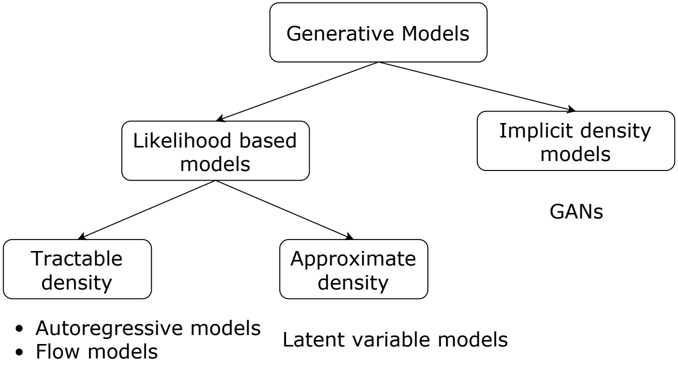
Agreement of different disentanglement metrics

Dataset = Noisy-dSprites

	(A)	(B)	(C)	(D)	(E)	(F)
BetaVAE Score (A)	100	80	44	41	46	37
FactorVAE Score (B)	80	100	49	52	25	38
MIG (C)	44	49	100	76	6	42
DCI Disentanglement (D)	41	52	76	100	-8	38
Modularity (E)	46	25	6	-8	100	13
SAP (F)	37	38	42	38	13	100

Locatello F. et al. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations, 2018

Generative models zoo



Likelihood based models

Is likelihood a good measure of model quality?

Poor likelihood
Great samples

$$p_1(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{N}(\mathbf{x}|\mathbf{x}_i, \epsilon \mathbf{I})$$

For small ϵ this model will generate samples with great quality, but likelihood will be very poor.

Great likelihood
Poor samples

$$p_2(\mathbf{x}) = 0.01 p(\mathbf{x}) + 0.99 p_{\text{noise}}(\mathbf{x})$$

$$\log\left[0.01 p(\mathbf{x}) + 0.99 p_{\text{noise}}(\mathbf{x})\right] \geq$$
$$\geq \log\left[0.01 p(\mathbf{x})\right] = \log p(\mathbf{x}) - \log 100$$

Noisy irrelevant samples, but for high dimensions $\log p(\mathbf{x})$ becomes proportional to m.

Theis L., Oord A., Bethge M. A note on the evaluation of generative models, 2015

Likelihood-free learning

- Likelihood is not a perfect measure quality measure for generative model.
- Likelihood could be intractable.

Where did we start

We would like to approximate true data distribution $\pi(\mathbf{x})$. Instead of searching true $\pi(\mathbf{x})$ over all probability distributions, learn function approximation $p(\mathbf{x}|\boldsymbol{\theta}) \approx \pi(\mathbf{x})$.

Imagine we have two sets of samples

- $\mathcal{S}_1 = \{\mathbf{x}_i\}_{i=1}^{n_1} \sim \pi(\mathbf{x})$ – real samples;
- $\mathcal{S}_2 = \{\mathbf{x}_i\}_{i=1}^{n_2} \sim p(\mathbf{x}|\boldsymbol{\theta})$ – generated (or fake) samples.

Two sample test

$$H_0 : \pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta}), \quad H_1 : \pi(\mathbf{x}) \neq p(\mathbf{x}|\boldsymbol{\theta})$$

Define test statistic $T(\mathcal{S}_1, \mathcal{S}_2)$. The test statistic is likelihood free. If $T(\mathcal{S}_1, \mathcal{S}_2) < \alpha$, then accept H_0, else reject it.

Likelihood-free learning

Two sample test

$$H_0 : \pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta}), \quad H_1 : \pi(\mathbf{x}) \neq p(\mathbf{x}|\boldsymbol{\theta})$$

Desired behaviour

- $p(\mathbf{x}|\boldsymbol{\theta})$ minimizes the value of test statistic $T(\mathcal{S}_1, \mathcal{S}_2)$.
- It is hard to find an appropriate test statistic in high dimensions. $T(\mathcal{S}_1, \mathcal{S}_2)$ could be learnable.

GAN objective

$$\min_G \max_D V(G, D) = \min_G \max_D \left[\mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(G(\mathbf{z})))\right]$$

- **Generator:** generative model $\mathbf{x} = G(\mathbf{z})$, which makes generated sample more realistic.
- **Discriminator:** a classifier $D(\mathbf{x}) \in [0, 1]$, which distinguishes real samples from generated samples.

Goodfellow I. J. et al. Generative Adversarial Networks, 2014

Vanilla GAN optimality

Theorem

The minimax game

$$\min_G \max_D V(G, D) = \min_G \max_D \left[\mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(G(\mathbf{z})))\right]$$

has the global optimum $\pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta})$, in this case $D^*(\mathbf{x}) = 0.5$.

Proof (fixed G)

$$\begin{aligned} V(G, D) &= \mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\theta})} \log(1 - D(\mathbf{x})) \\ &= \int \underbrace{[\pi(\mathbf{x}) \log D(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta}) \log(1 - D(\mathbf{x})]}_{y(D)} d\mathbf{x} \end{aligned}$$

$$\frac{dy(D)}{dD} = \frac{\pi(\mathbf{x})}{D(\mathbf{x})} - \frac{p(\mathbf{x}|\boldsymbol{\theta})}{1 - D(\mathbf{x})} = 0 \quad \Rightarrow \quad D^*(\mathbf{x}) = \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}$$

Goodfellow I. J. et al. Generative Adversarial Networks, 2014

Vanilla GAN optimality

Proof confitnued (fixed $D = D^*$)

$$V(G, D^*) = \mathbb{E}_{\pi(\mathbf{x})} \log \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})} + \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\theta})} \log \frac{p(\mathbf{x}|\boldsymbol{\theta})}{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}$$

$$= KL\left(\pi(\mathbf{x}) || \frac{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}{2}\right) + KL\left(p(\mathbf{x}|\boldsymbol{\theta}) || \frac{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}{2}\right) - 2\log 2$$

$$= 2JSD(\pi(\mathbf{x}) || p(\mathbf{x}|\boldsymbol{\theta})) - 2\log 2.$$

Jensen-Shannon divergence (symmetric KL divergence)

$$JSD(\pi(\mathbf{x}) || p(\mathbf{x}|\boldsymbol{\theta})) = \frac{1}{2}\left[KL\left(\pi(\mathbf{x}) || \frac{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}{2}\right) + KL\left(p(\mathbf{x}|\boldsymbol{\theta}) || \frac{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}{2}\right)\right]$$

Could be used as a distance measure!

$$V(G^*, D^*) = -2\log 2, \quad \pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta}).$$

Goodfellow I. J. et al. Generative Adversarial Networks, 2014

Vanilla GAN optimality

Theorem

The minimax game

$$\min_G \max_D V(G, D) = \min_G \max_D \left[\mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(G(\mathbf{z})))\right]$$

has the global optimum $\pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta})$, in this case $D^*(\mathbf{x}) = 0.5$.

Proof

for fixed G:

$$D^*(\mathbf{x}) = \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}) + p(\mathbf{x}|\boldsymbol{\theta})}$$

for fixed $D = D^*$:

$$\min_G V(G, D^*) = \min_G \left[2JSD(\pi||p) - \log 4\right] = -\log 4, \quad \pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta}).$$

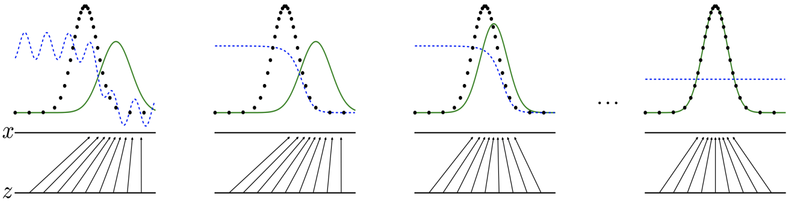
If the generator could be any function and the discriminator is optimal at every step, then the generator is guaranteed to converge to the data distribution.

Goodfellow I. J. et al. Generative Adversarial Networks, 2014

Vanilla GAN

Objective

$$\min_G \max_D V(G, D) = \min_G \max_D \left[\mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(G(\mathbf{z}))) \right]$$



(a) (b) (c) (d)

- Generator updates are made in parameter space.
- Discriminator is not optimal at every step.
- Generator and discriminator loss keeps oscillating during GAN training.

Goodfellow I. J. et al. Generative Adversarial Networks, 2014

Summary

- Majority of disentanglement learning models use heuristic objective or regularizers to achieve the goal, but the task itself could not be solved without good inductive bias.
- Likelihood is not a perfect criteria to measure quality of generative model.
- Adversarial learning suggest to solve minimax problem to match the distributions.
- Vanilla GAN tries to optimize Jensen-Shannon divergence (in theory).