Latent Variable Models (Part 3)

Shen Ting Ang

19 Sep 2019

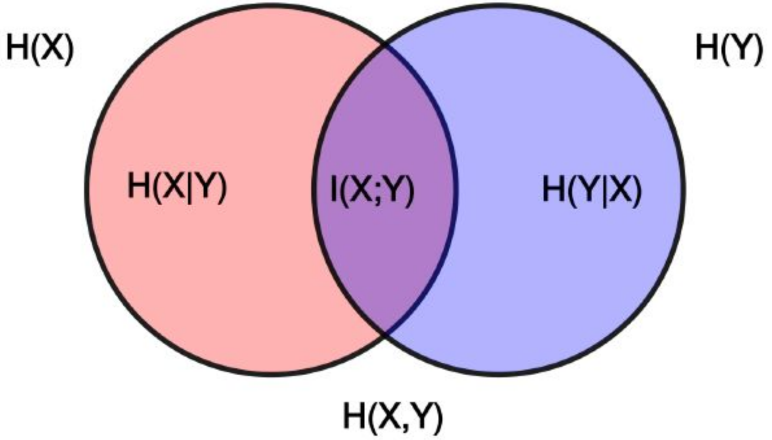
Outline

- Warm-up on Variational Inference
  - ~~Recap~~
  - ~~An importance sampling view~~
  - **Variational Mutual Information Estimation/Maximization**
  - Variational Dequantization
- Improving VAEs
  - ~~Reducing variational gap~~
  - ~~More flexible decoder & posterior collapse problem~~
  - More expressive architectures

Mutual Information

- Mutual Information between two random variables X, Y: I(X;Y) is defined as:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

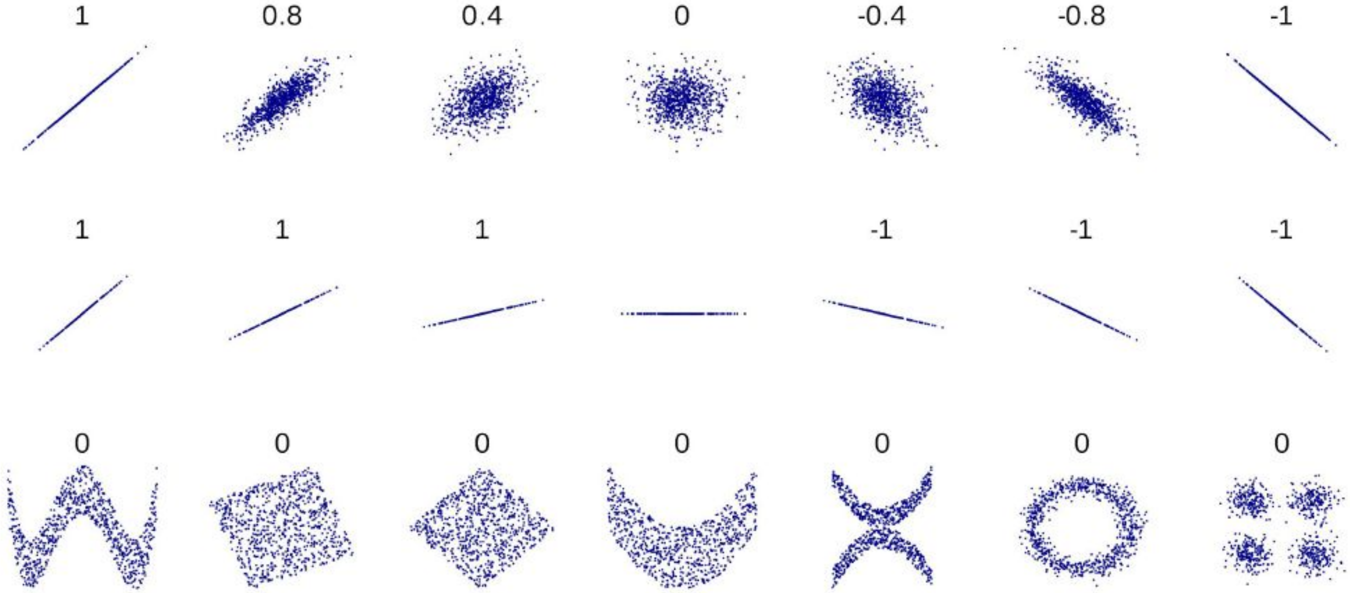


Mutual Information and Dependency

- Mutual Information: General way to measure ***dependency*** between two random variables
- Don’t we already have correlation? Why dependency over correlation?

Correlation vs Dependency

Does lack of correlation imply lack of dependence? No



Mutual Information

- Useful in a lot of settings where one wants to maximize dependency between two variables or estimate their dependencies:
  - Variational Information Maximisation for Intrinsically Motivated Reinforcement Learning
  - InfoGan
  - Contrastive Predictive Coding
    (Code: https://github.com/jefflai108/Contrastive-Predictive-Coding-PyTorch)

Outline

- Warm-up on Variational Inference
    - ~~Recap~~
    - ~~An importance sampling view~~
    - ~~Variational Mutual Information Estimation/Maximization~~
    - **Variational Dequantization**
- Improving VAEs
    - ~~Reducing variational gap~~
    - ~~More flexible decoder & posterior collapse problem~~
    - More expressive architectures

Uniform Dequantization (Recap)

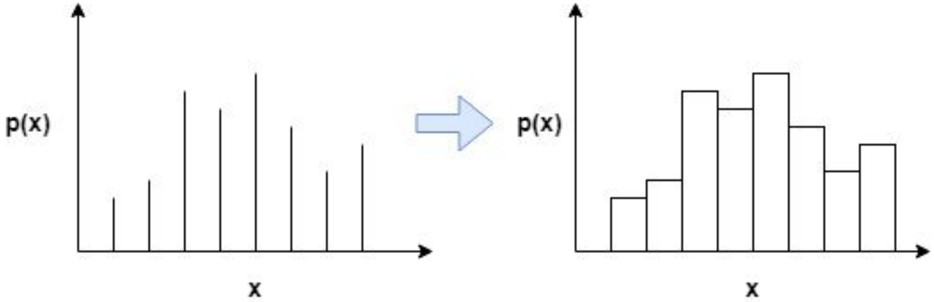
- Idea: Add noise to data
  - E.g. Image data: $x \in \{0, 1, 2, \dots, 255\}$
  - Add noise u~Uniform $[0,1)^D$

$$
\begin{aligned}
\mathbb{E}_{\mathbf{y}\sim p_{\text{data}}}\left[\log p_{\text{model}}(\mathbf{y})\right] &= \sum_{\mathbf{x}} P_{\text{data}}(\mathbf{x}) \int_{[0,1)^D} \log p_{\text{model}}(\mathbf{x}+\mathbf{u})\, d\mathbf{u} \\
&\leq \sum_{\mathbf{x}} P_{\text{data}}(\mathbf{x}) \log \int_{[0,1)^D} p_{\text{model}}(\mathbf{x}+\mathbf{u})\, d\mathbf{u} \\
&= \mathbb{E}_{\mathbf{x}\sim P_{\text{data}}}\left[\log P_{\text{model}}(\mathbf{x})\right]
\end{aligned}
$$

[Theis, Oord, Bethge, 2016]

Uniform Dequantization (Recap)

- Idea: Add noise to data
  - E.g. Image data: $x \in \{0, 1, 2, \dots, 255\}$
  - Add noise u~Uniform $[0,1)^D$



Problems:

- P_{model} assigns uniform density to unit hypercubes - unnatural!
- Neural networks are usually smooth functions

Variable Dequantization

Idea: Learn noise q using Variational Inference

$$
\begin{aligned}
\mathbb{E}_{\mathbf{x}\sim P_{\text{data}}}\left[\log P_{\text{model}}(\mathbf{x})\right] &= \mathbb{E}_{\mathbf{x}\sim P_{\text{data}}}\left[\log \int_{[0,1)^D} q(\mathbf{u}|\mathbf{x}) \frac{p_{\text{model}}(\mathbf{x}+\mathbf{u})}{q(\mathbf{u}|\mathbf{x})}\, d\mathbf{u}\right] \\
&\geq \mathbb{E}_{\mathbf{x}\sim P_{\text{data}}}\left[\int_{[0,1)^D} q(\mathbf{u}|\mathbf{x}) \log \frac{p_{\text{model}}(\mathbf{x}+\mathbf{u})}{q(\mathbf{u}|\mathbf{x})}\, d\mathbf{u}\right] \\
&= \mathbb{E}_{\mathbf{x}\sim P_{\text{data}}}\mathbb{E}_{\mathbf{u}\sim q(\cdot|\mathbf{x})}\left[\log \frac{p_{\text{model}}(\mathbf{x}+\mathbf{u})}{q(\mathbf{u}|\mathbf{x})}\right]
\end{aligned}
$$

[Ho et al., 2019]

Variable Dequantization

Intuition:

- Learn easy to fit dequantization noise
- “Find points in the interval which is easy for model to maximize”
- $u \sim q(u|x)$ is analogous to VAE

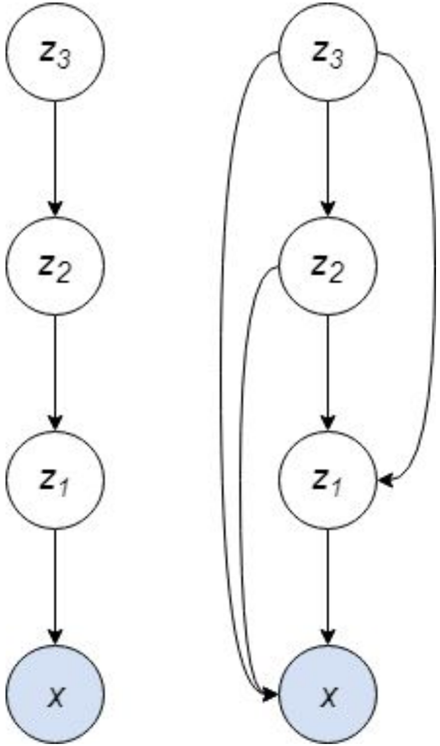
How to train: Train both models jointly.

Outline

- Warm-up on Variational Inference
  - ~~Recap~~
  - ~~An importance sampling view~~
  - ~~Variational Mutual Information Estimation/Maximization~~
  - ~~Variational Dequantization~~
- Improving VAEs
  - ~~Reducing variational gap~~
  - ~~More flexible decoder & posterior collapse problem~~
  - **More expressive architectures**

Hierarchical Latent Variables

Idea: Chain latent variables (Markov Chain or autoregressive)



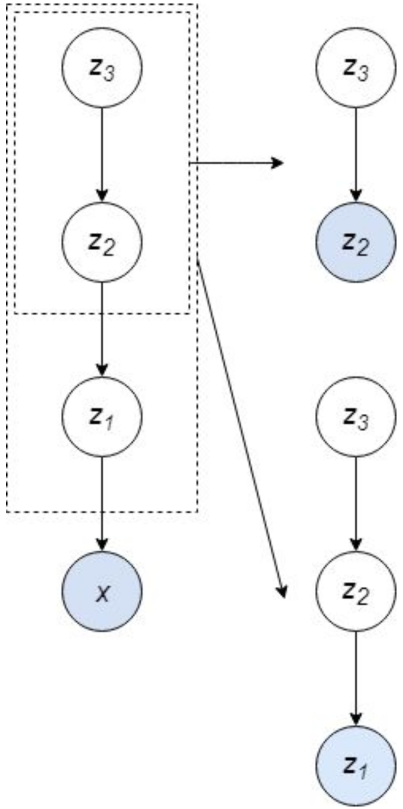
$$p(x, z) = p(x|z)p(z)$$

$$p(x, z_{1:L}) = p(x|z_{1:L}) \left(\prod_{i=1}^{L-1} p(z_i|z_{i+1:L}) \right) p(z_L)$$

Hierarchical Latent Variables

Idea: “Nested” VAEs

- More latent variables -> More powerful distributions
- More modelling capacity



Training multiple latent variables

Idea: Treat latent variables as one latent variable

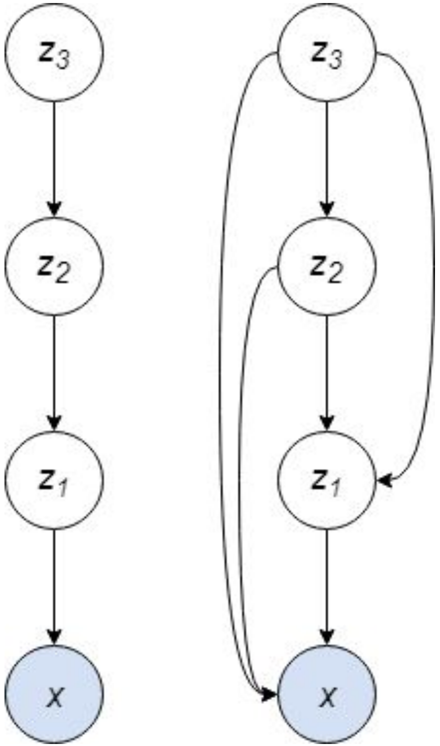
Generation:

$$p(x, z_{1:L}) = p(x|z_{1:L}) \left(\prod_{i=1}^{L-1} p(z_i|z_{i+1:L}) \right) p(z_L)$$

Variational Lower Bound:

$$\log p(x) \geq \mathbb{E}_{z_{1:L} \sim q(z_{1:L}|x)} \left[\log \frac{\boxed{p(x, z_{1:L})}}{q(z_{1:L}|x)} \right]$$

- Evaluating/Differentiating p(x,z) is fast
- Deepest models are about 20 latent variables, so slower sampling isn't so much of an issue (as compared to sampling thousands)



Inference networks for hierarchical models

- $q(z_{1:L}|x)$ should be as flexible as possible, yet fast to sample for fast training
- Examples:
  - IAF-VAE (Kingma et al. 2016) - IAF for each z, stitched together autoregressively over layers
  - Bi-directional Inference Variational Autoencoder (BIVA) (Maaløe et al. 2019) - uses autoregressive flows over 1:L; Very effective, SOTA on many benchmarks
  - Autoregressive structure is over layers (not dimensions of data), hence sampling speed is still acceptable.