Deep Generative Models

5. Latent variable models

- 국가수리과학연구소 산업수학혁신센터 김민중

Recap

- Autoregressive models:
  - Chain rule-based factorization is fully general
  - Compact representation via conditional independence and/or neural parameterizations
- Autoregressive models Pros
  - Easy to evaluate likelihoods
  - Easy to train
- Autoregressive models Cons
  - Requires an ordering
  - Generation is sequential
  - Cannot learn features in an unsupervised way

Plan for this lecture

- Latent Variable Models
  - Mixture models
  - Variational autoencoder
  - Variational inference and learning

Latent Variable Models: Assumption

- Observable variables $\boldsymbol{x} \in \mathbb{R}^d$
- Latent variables $\boldsymbol{z} \in \mathbb{R}^h$ (unobservable)

$$p_{data}(\boldsymbol{x}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z})$$

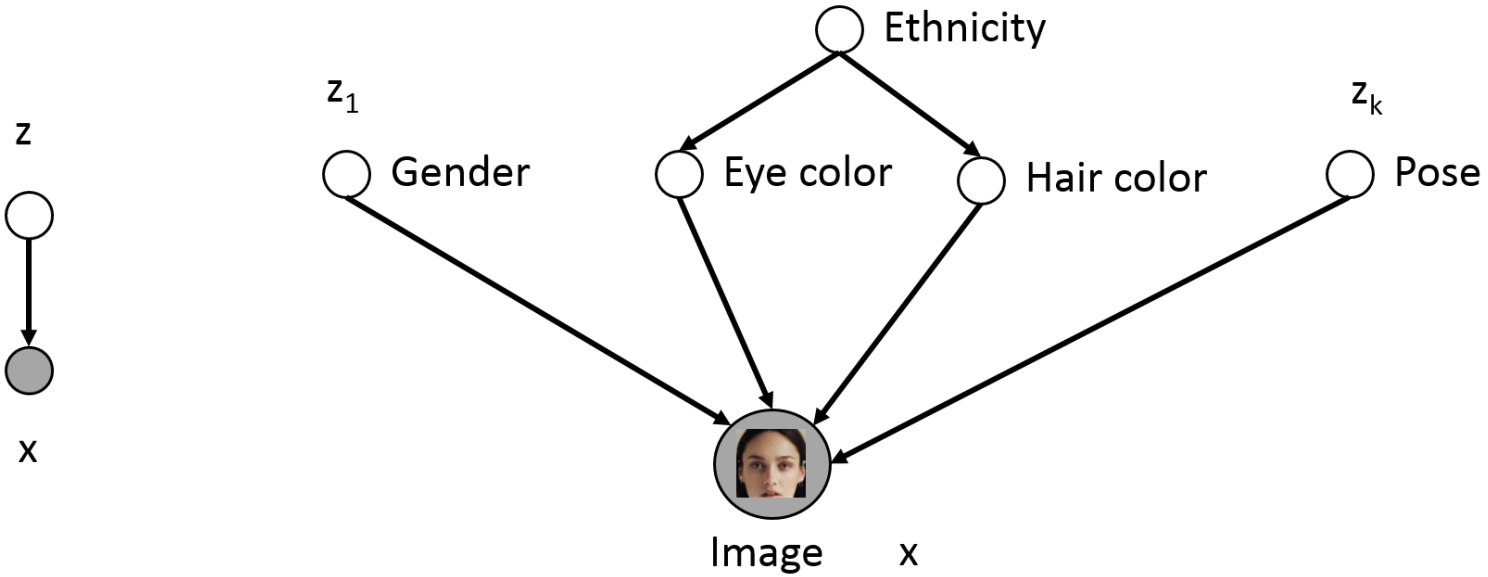
$$or \qquad = \int p(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z}$$

Latent Variable Models: Motivation

- Lots of variability in images $\boldsymbol{x}$ due to gender, eye color, hair color, pose, etc.
- However, unless images are annotated, these factors of variation are not explicitly available (latent)
- Idea: explicitly model these factors using latent variables $\boldsymbol{z}$

Latent Variable Models

- Only shaded variables $\boldsymbol{x}$ are observed in the data (pixel values)
- Latent variables $\boldsymbol{z}$ correspond to high level features
  - If $\boldsymbol{z}$ chosen properly, $p(\boldsymbol{x}|\boldsymbol{z})$ could be much simpler than $\boldsymbol{p}(\boldsymbol{x})$
  - If we had trained this model, then we could identify features via $p(\boldsymbol{z}|\boldsymbol{x})$, e.g., $p(\mathrm{EyeColor} = \mathrm{Blue}|\boldsymbol{x})$
- Challenge: Very difficult to specify these conditionals by hand

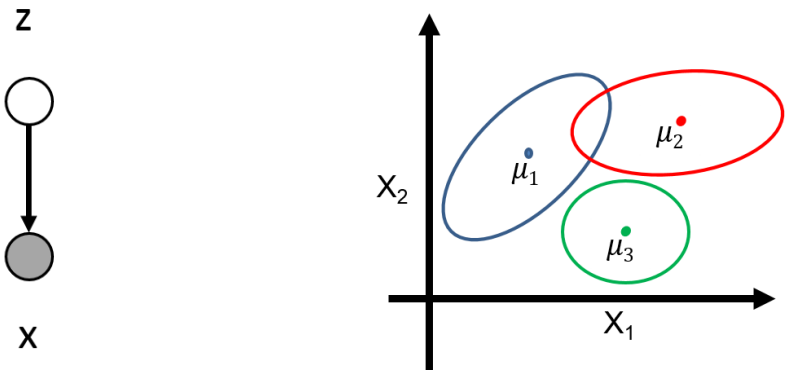


Why Latent Variable Models?

- Curse of Dimensionality
  - Sparse and peaky
- Model $p(\boldsymbol{x}, \boldsymbol{z})$ instead of $p(\boldsymbol{x})$
- Maximizing $p_\theta(\boldsymbol{x}, \boldsymbol{z})$ and Maximizing $p_\theta(\boldsymbol{x})$
  - Easy to sample a new data
  - Neural network can learn and model a mapping effectively. E.g., $p_\theta(\boldsymbol{x}|\boldsymbol{z})$

Mixture of Gaussians: a Shallow Latent Variable Model

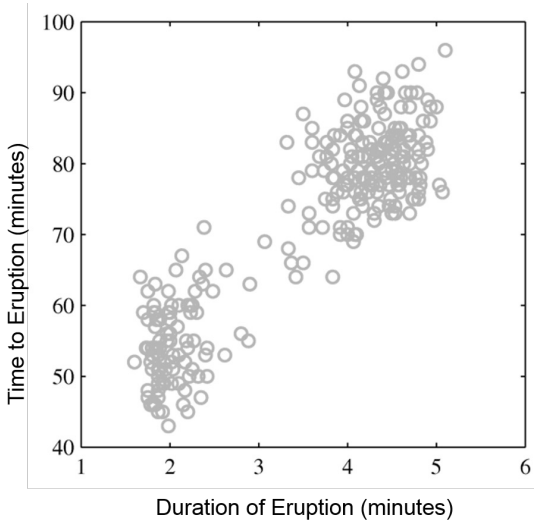
- Mixture of Gaussians. Bayes net: $z \rightarrow \boldsymbol{x}$
  - $z = Categorical(z|\gamma_1, \cdots, \gamma_K)$
  - $p(\boldsymbol{x}|z = k) = N(\boldsymbol{x}|\mu_k, \Sigma_k)$

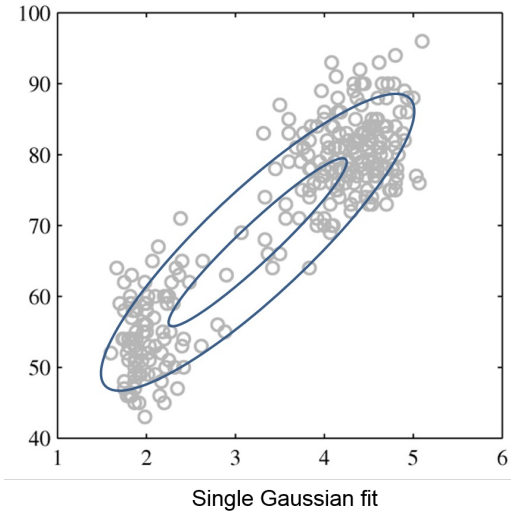


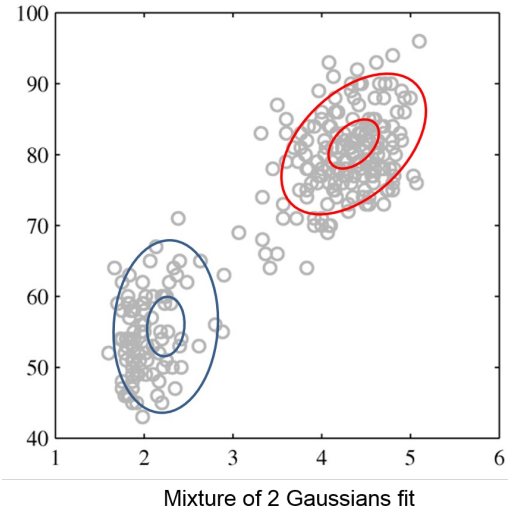
- Generative Process
  - Pick a mixture component k by sampling z
  - Generate a data point by sampling from that Gaussian

Mixture of Gaussians: a Shallow Latent Variable Model

- Mixture of Gaussians. Bayes net: $z \rightarrow \boldsymbol{x}$
  - $z = Categorical(z|\gamma_1, \cdots, \gamma_K)$
  - $p(\boldsymbol{x}|z = k) = N(\boldsymbol{x}|\mu_k, \Sigma_k)$



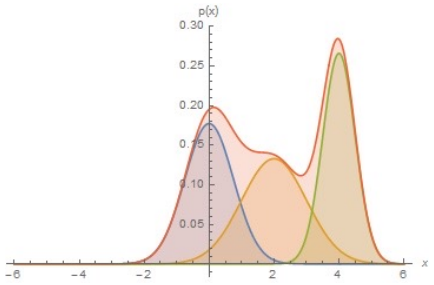




- Clustering: The posterior $p(z|\boldsymbol{x})$ identifies the mixture component

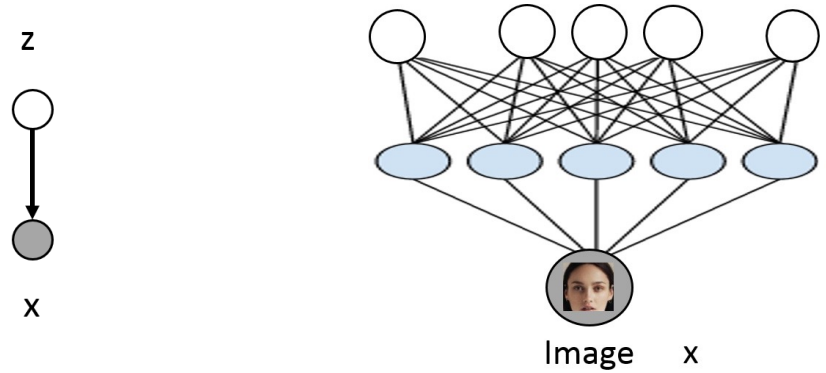
Mixture Models

- Alternative motivation: Combine simple models into a more complex and expressive one



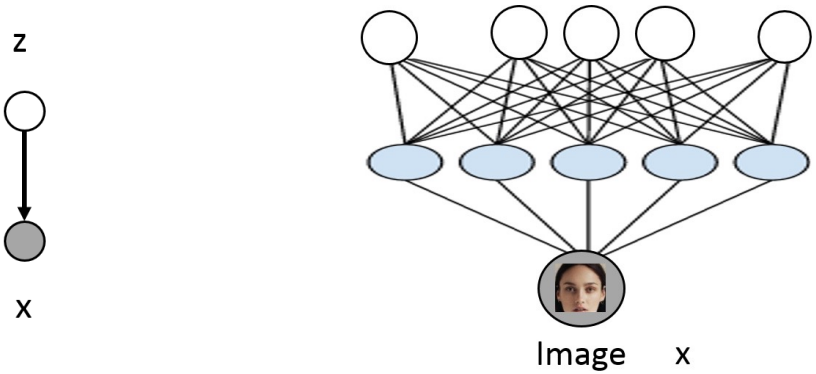
$$p(\boldsymbol{x}) = \sum_z p(\boldsymbol{x}, z) = \sum_z p(z)p(\boldsymbol{x}|z) = \sum_z p(z = k)N(\boldsymbol{x}|\mu_k, \Sigma_k)$$

Example: Variational Autoencoder



- A Mixture of an infinite number of Gaussians
  - $\boldsymbol{z} = N(\boldsymbol{z}|\boldsymbol{0}, \boldsymbol{I})$, $\boldsymbol{z} \in \mathbb{R}^h$
  - $p(\boldsymbol{x}|\boldsymbol{z}) = N(\boldsymbol{x}|\mu_\theta(\boldsymbol{z}), \Sigma_\theta(\boldsymbol{z}))$where $\mu_\theta, \Sigma_\theta$ are neural networks
  - Even though $p(\boldsymbol{x}|\boldsymbol{z})$ is simple, the marginal $p(\boldsymbol{x})$ is very complex/flexible
- Hope that after training, $\boldsymbol{z}$ will correspond to meaningful latent factors of variation (features)
- Unsupervised representation learning
- Features can be computed via $p(\boldsymbol{z}|\boldsymbol{x})$

Example: Variational Autoencoder



- A Mixture of an infinite number of Gaussians
  - $\boldsymbol{z} = N(\boldsymbol{z}|\boldsymbol{0}, \boldsymbol{I}),\ \boldsymbol{z} \in \mathbb{R}^h$
  - $p(\boldsymbol{x}|\boldsymbol{z}) = N(\boldsymbol{x}|\mu_\theta(\boldsymbol{z}), \Sigma_\theta(\boldsymbol{z}))$where $\mu_\theta, \Sigma_\theta$ are neural networks
    - $\mu_\theta(\boldsymbol{z}) = \sigma(A\boldsymbol{z} + \boldsymbol{c}) = (\sigma(\boldsymbol{a}_1^T\boldsymbol{z} + c_1), \sigma(\boldsymbol{a}_2^T\boldsymbol{z} + c_2))$
    - $\Sigma_\theta(\boldsymbol{z}) = diag(\exp\sigma(B\boldsymbol{z} + \boldsymbol{d})) = \begin{pmatrix} \exp\sigma(\boldsymbol{b}_1^T\boldsymbol{z} + d_1) & 0 \\ 0 & \exp\sigma(\boldsymbol{b}_2^T\boldsymbol{z} + d_2) \end{pmatrix}$
    - $\theta = (A, B, \boldsymbol{c}, \boldsymbol{d})$

Recap

- Latent Variable Models
  - Allow us to define complex models $p(\boldsymbol{x})$ in terms of simpler building blocks $p(\boldsymbol{x}|\boldsymbol{z})$
  - Natural for unsupervised learning tasks (clustering, unsupervised representation learning, etc.)
  - No free lunch: much more difficult to learn compared to fully observed, autoregressive models

Marginal Likelihood

- Let X denote observed random variables, and Z the unobserved hidden or latent variables
- Suppose we have a model for the joint distribution

$$p_\theta(X, Z)$$

- What is the probability $p_\theta(X = \bar{x})$ of observing a training data point $\bar{x}$?

$$p_\theta(\bar{x}) = \sum_z p_\theta(X = \bar{x}, Z = z) = \sum_z p_\theta(\bar{x}, z)$$

Partially observed data

- Suppose we have a model for the joint distribution
$$p_\theta(X, Z)$$
- We have a dataset D, where for each datapoint the X variables are observed (e.g., pixel values) and the latent variables Z are never observed. $D = \{\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(N)}\}$
- Maximum likelihood learning
$$\log \prod_{\boldsymbol{x} \in D} p_\theta(\boldsymbol{x}) = \sum_{\boldsymbol{x} \in D} \log p_\theta(\boldsymbol{x}) = \sum_{\boldsymbol{x} \in D} \log \sum_{\boldsymbol{z}} p_\theta(\boldsymbol{x}, \boldsymbol{z})$$
- Evaluating $\log \sum_{\boldsymbol{z}} p_\theta(\boldsymbol{x}, \boldsymbol{z})$ can be intractable. For continuous variables, $\log \int p_\theta(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z}$ is often intractable
- Gradients ∇_θ also hard to compute
- **Need approximations**: One gradient evaluation per training data point $\boldsymbol{x} \in D$, so approximation needs to be cheap

Attempt: Importance Sampling

- Likelihood function $p_\theta(X)$ for partially observed data is hard to compute:

$$p_\theta(\boldsymbol{x}) = \sum_{\boldsymbol{z}\in\mathcal{Z}} p_\theta(\boldsymbol{x},\boldsymbol{z}) = \sum_{\boldsymbol{z}\in\mathcal{Z}} \frac{q(\boldsymbol{z})}{q(\boldsymbol{z})} p_\theta(\boldsymbol{x},\boldsymbol{z}) = E_{\boldsymbol{z}\sim q(\boldsymbol{z})}\left[\frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z})}\right]$$

- Monte Carlo to the rescue:
  - Sample $\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \cdots, \boldsymbol{z}^{(K)}$ from $q(\boldsymbol{z})$
  - Approximate expectation with sample average

$$p_\theta(\boldsymbol{x}) \approx \frac{1}{K}\sum_{j=1}^{K} \frac{p_\theta\left(\boldsymbol{x},\boldsymbol{z}^{(j)}\right)}{q(\boldsymbol{z}^{(j)})}$$

  - What is a good choice for $q(\boldsymbol{z})$? Intuitively, frequently sample $\boldsymbol{z}$ that are likely given $\boldsymbol{x}$ under $p_\theta(\boldsymbol{x},\boldsymbol{z})$.

Estimating log-likelihoods

- Monte Carlo to the rescue:
  - Sample $\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \cdots, \boldsymbol{z}^{(K)}$ from $q(\boldsymbol{z})$
  - Approximate expectation with sample average(unbiased)

$$p_\theta(\boldsymbol{x}) \approx \frac{1}{K}\sum_{j=1}^{K}\frac{p_\theta\left(\boldsymbol{x}, \boldsymbol{z}^{(j)}\right)}{q(\boldsymbol{z}^{(j)})}$$

  - Recall that for training, we need the log-likelihood $\log p_\theta(\boldsymbol{x})$

$$\log p_\theta(\boldsymbol{x}) \approx \log\left(\frac{1}{K}\sum_{j=1}^{K}\frac{p_\theta\left(\boldsymbol{x}, \boldsymbol{z}^{(j)}\right)}{q(\boldsymbol{z}^{(j)})}\right) \approx \log\left(\frac{p_\theta(\boldsymbol{x}, \boldsymbol{z}')}{q(\boldsymbol{z}')}\right)$$

  - However, it is clear that

$$\log\left(E_{\boldsymbol{z}\sim q(\boldsymbol{z})}\left[\frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q(\boldsymbol{z})}\right]\right) \neq E_{\boldsymbol{z}\sim q(\boldsymbol{z})}\left[\log\left(\frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q(\boldsymbol{z})}\right)\right]$$

Evidence Lower Bound

- Log-Likelihood function

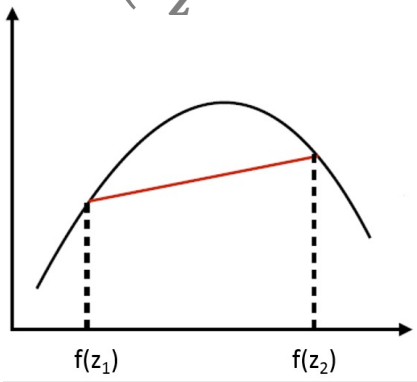
$$\log \sum_{\boldsymbol{z} \in \mathcal{Z}} p_\theta(\boldsymbol{x}, \boldsymbol{z}) = \log \sum_{\boldsymbol{z} \in \mathcal{Z}} \frac{q(\boldsymbol{z})}{q(\boldsymbol{z})} p_\theta(\boldsymbol{x}, \boldsymbol{z}) = \log \left(E_{\boldsymbol{z} \sim q(\boldsymbol{z})} \left[\frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q(\boldsymbol{z})} \right] \right)$$

- $\log$ is a concave function

$$\log(px + (1-p)x') \geq p \log(x) + (1-p) \log(x')$$

- By Jensen Inequality(for concave function),

$$\log\left(E_{\boldsymbol{z} \sim q(\boldsymbol{z})}[f(\boldsymbol{z})]\right) = \log\left(\sum_{\boldsymbol{z}} q(\boldsymbol{z}) f(\boldsymbol{z})\right) \geq \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log f(\boldsymbol{z})$$



Evidence Lower Bound

- Log-Likelihood function

$$\log \sum_{\boldsymbol{z}\in\mathcal{Z}} p_\theta(\boldsymbol{x},\boldsymbol{z}) = \log \sum_{\boldsymbol{z}\in\mathcal{Z}} \frac{q(\boldsymbol{z})}{q(\boldsymbol{z})} p_\theta(\boldsymbol{x},\boldsymbol{z}) = \log\left(E_{\boldsymbol{z}\sim q(\boldsymbol{z})}\left[\frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z})}\right]\right)$$

- $\log$ is a concave function

$$\log(px + (1-p)x') \geq p\log(x) + (1-p)\log(x')$$

- By Jensen Inequality(for concave function),

$$\log\left(E_{\boldsymbol{z}\sim q(\boldsymbol{z})}[f(\boldsymbol{z})]\right) = \log\left(\sum_{\boldsymbol{z}} q(\boldsymbol{z}) f(\boldsymbol{z})\right) \geq \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log f(\boldsymbol{z})$$

- Choosing $f(\boldsymbol{z}) = \frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z})}$

$$\log\left(E_{\boldsymbol{z}\sim q(\boldsymbol{z})}\left[\frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z})}\right]\right) \geq \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log \frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z})}$$

Variational inference

- Suppose $q(\boldsymbol{z})$ is any probability distribution over the hidden variables
- Evidence lower bound (ELBO) holds for any $q(\boldsymbol{z})$

$$\log p_\theta(\boldsymbol{x}) \geq \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q(\boldsymbol{z})}$$

$$= \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log p_\theta(\boldsymbol{x}, \boldsymbol{z}) - \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log q(\boldsymbol{z})$$

$$= \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log p_\theta(\boldsymbol{x}, \boldsymbol{z}) + H(q)$$

- Claim: Equality holds if $q(\boldsymbol{z}) = p_\theta(\boldsymbol{z}|\boldsymbol{x})$

Variational inference

- We derived this lower bound that holds holds for any choice of $q(\boldsymbol{z})$

$$\log p_\theta(\boldsymbol{x}) \geq \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{q(\boldsymbol{z})}$$

- If $q(\boldsymbol{z}) = p_\theta(\boldsymbol{z}|\boldsymbol{x})$, then the bound becomes

$$\sum_{\boldsymbol{z}} p_\theta(\boldsymbol{z}|\boldsymbol{x}) \log \frac{p_\theta(\boldsymbol{x}, \boldsymbol{z})}{p_\theta(\boldsymbol{z}|\boldsymbol{x})} = \sum_{\boldsymbol{z}} p_\theta(\boldsymbol{z}|\boldsymbol{x}) \log \frac{p_\theta(\boldsymbol{z}|\boldsymbol{x}) p_\theta(\boldsymbol{x})}{p_\theta(\boldsymbol{z}|\boldsymbol{x})}$$

$$= \sum_{\boldsymbol{z}} p_\theta(\boldsymbol{z}|\boldsymbol{x}) \log p_\theta(\boldsymbol{x}) = \log p_\theta(\boldsymbol{x}) \sum_{\boldsymbol{z}} p_\theta(\boldsymbol{z}|\boldsymbol{x}) = \log p_\theta(\boldsymbol{x})$$

- What if the posterior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ is intractable to compute? How loose is the bound?

Variational inference(continued)

- Suppose $q(\boldsymbol{z})$ is any probability distribution over the hidden variables. A little bit of algebra reveals

$$D\big(q(\boldsymbol{z}) \parallel p_\theta(\boldsymbol{z}|\boldsymbol{x})\big) = -\sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log p_\theta(\boldsymbol{x}, \boldsymbol{z}) + \log p_\theta(\boldsymbol{x}) - H(q) \geq 0$$

- Rearranging, we re-derived the Evidence lower bound (ELBO)

$$\log p_\theta(\boldsymbol{x}) \geq \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \log p_\theta(\boldsymbol{x}, \boldsymbol{z}) + H(q)$$

- Equality holds if $q(\boldsymbol{z}) = p_\theta(\boldsymbol{z}|\boldsymbol{x})$ because $D\big(q(\boldsymbol{z}) \parallel p_\theta(\boldsymbol{z}|\boldsymbol{x})\big) = 0$
- In general, $\log p_\theta(\boldsymbol{x}) = \mathrm{ELBO} + D\big(q(\boldsymbol{z}) \parallel p_\theta(\boldsymbol{z}|\boldsymbol{x})\big)$

The Evidence Lower bound

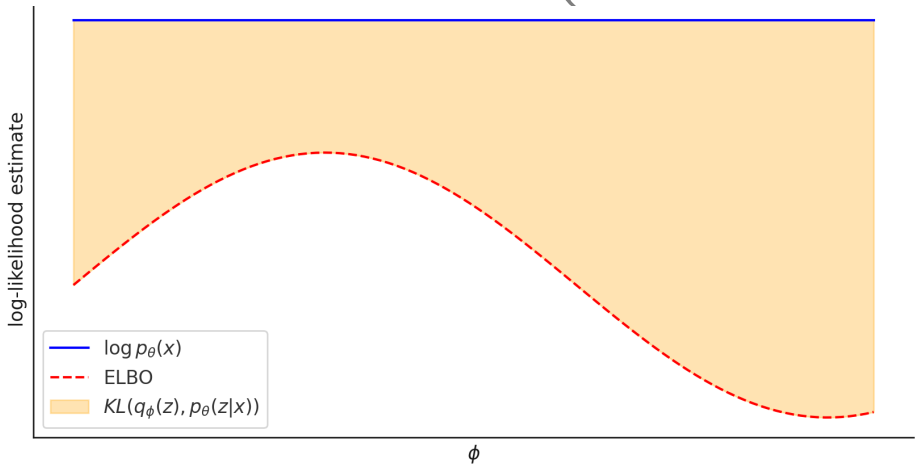
- What if the posterior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ is intractable to compute?
- Suppose $q_\phi(\boldsymbol{z})$ is a (tractable) probability distribution over the hidden variables parameterized by ϕ (variational parameters)
  - For example, a Gaussian with mean and covariance specified by ϕ

$$q_\phi(\boldsymbol{z}) = N\left(\boldsymbol{z}\middle|\boldsymbol{\mu}_\phi, \Sigma_\phi\right)$$

- Variational inference: pick ϕ so that $q_\phi(\boldsymbol{z})$ is as close as possible to $p_\theta(\boldsymbol{z}|\boldsymbol{x})$

The Evidence Lower bound

$$\log p_\theta(\boldsymbol{x}) = \text{ELBO} + D\left(q_\phi(\boldsymbol{z}) \parallel p_\theta(\boldsymbol{z}|\boldsymbol{x})\right)$$



- The better $q_\phi(\boldsymbol{z})$ can approximate the posterior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$, the smaller $D\left(q_\phi(\boldsymbol{z}) \parallel p_\theta(\boldsymbol{z}|\boldsymbol{x})\right)$ we can achieve, the closer ELBO will be to $\log p_\theta(\boldsymbol{x})$
- Next: jointly optimize over θ and ϕ to maximize the ELBO over a dataset

Summary

- Latent Variable Models Pros:
  - Easy to build flexible models
  - Suitable for unsupervised learning
- Latent Variable Models Cons
  - Hard to evaluate likelihoods
  - Hard to train via maximum-likelihood
  - Fundamentally, the challenge is that posterior inference $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ is hard. Typically requires variational approximations
- Alternative: give up on KL-divergence and likelihood (GANs)

Thanks