Machine Learning for Graphs and Sequential Data

Deep Generative Models - Variational Inference

Lecturer: Prof. Dr. Stephan Günnemann

daml.in.tum.de

Summer Term 2020

Data Analytics and
Machine Learning

Roadmap

- Chapter: Deep Generative Models
  1. Introduction
  2. Normalizing Flows
  3. **Variational Inference**
     - **Latent variable models**
     - Maximization using lower bounds
     - Optimizing the ELBO
     - Variational Autoencoders
  4. Generative Adversarial Networks

Latent Variable Models (LVMs)

- We want to model a probability distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x})$

- The data $\boldsymbol{x}$ is high-dimensional, but we can often describe it using only few latent factors $\boldsymbol{z}$

- For example, an image can be compactly represented by considering
  - Objects in the scene, their locations & colors
  - Lighting
  - Viewing angle
  - …

- We can exploit this low-dimensional latent structure in our probabilistic model

Latent Variable Models (LVMs)

$z \sim \mathcal{N}(3, 0.5)$

$z = 2.9$

$W = \begin{bmatrix} 2 \\ 1 \\ 7 \end{bmatrix}$

$x \sim \mathcal{N}\left(W \cdot z, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)$

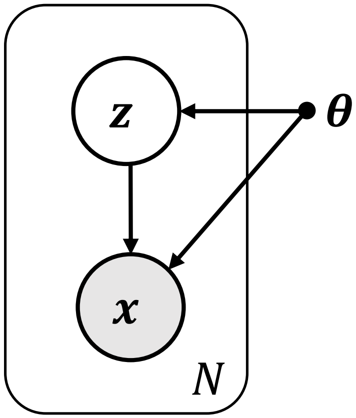
- LVM defines a two-step process for generating the data
  1. Generate (i.e. sample) the latent variable $\boldsymbol{z}$

     $$\boldsymbol{z} \sim p_{\boldsymbol{\theta}}(\boldsymbol{z})$$

     $x = \begin{bmatrix} 6.3 \\ 3 \\ 20.4 \end{bmatrix}$

  2. Generate the data $\boldsymbol{x}$ conditional on $\boldsymbol{z}$

     $$\boldsymbol{x} \sim p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$$



- The above procedure defines the joint distribution

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

- Marginal likelihood

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z} = \int p_{\boldsymbol{\theta}}(\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) d\boldsymbol{z} = \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{\theta}}(\boldsymbol{z})}[p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})]$$

Example: Gaussian Mixture Model

$\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$

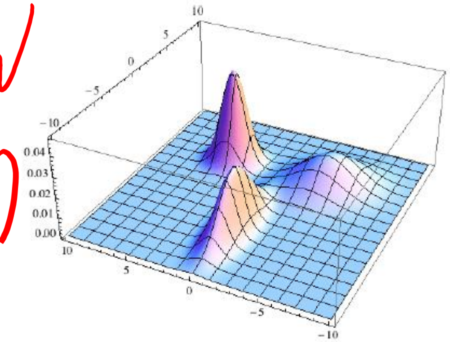
$Z \sim CAT(\pi)$

$z = 3$

- Gaussian mixture model
  $$p(z = k) = \pi_k$$
  $$p(\boldsymbol{x}|z = k) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$
  $$p(\boldsymbol{x}) = \textstyle\sum_k \pi_k \, \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$
  - Summation instead of integration since z is discrete

$x \sim \mathcal{N}(\mu_3, \Sigma_3)$

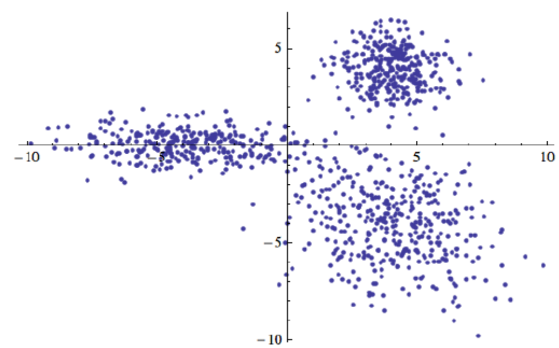
$x = \begin{bmatrix} 0.3 \\ 1.1 \end{bmatrix}$ $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$



(a) A probability distribution on $\mathbb{R}^2$. [1]

- Parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K, \pi_1, \ldots, \pi_K)$
  - component means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$, weights π_k

- Main idea of a LVM
  - The conditional distribution $p(\boldsymbol{x}|\boldsymbol{z})$ is “simple”
  - The marginal distribution $p(\boldsymbol{x})$ is “complex”



(b) Data sampled from this distribution.

Tasks in LVMs

- Inference: Given a sample $\boldsymbol{x}$, find the posterior distribution over $\boldsymbol{z}$
  - This can be viewed as "extracting" the latent features

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

- Learning: Given a dataset $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ (usually consisting of i.i.d. samples), find the parameters $\boldsymbol{\theta}$ that best explain the data
  - Typically done by maximizing the marginal log-likelihood

$$\max_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{X}) = \max_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$$

i.i.d. assumption

Maximum Likelihood Estimation in LVMs

- For simplicity, we first assume that we want to maximize the marginal log-likelihood for a single sample $\boldsymbol{x}$. We will handle the general case later

$$\begin{aligned}\max_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) &= \max_{\boldsymbol{\theta}} \log\left(\int p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})d\boldsymbol{z}\right) \\ &= \max_{\boldsymbol{\theta}} \underbrace{\log\left(\int p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p_{\theta}(\boldsymbol{z})d\boldsymbol{z}\right)}_{f(\boldsymbol{\theta})} \\ &= \max_{\boldsymbol{\theta}} f(\boldsymbol{\theta})\end{aligned}$$

- In general, the integral $\int p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})d\boldsymbol{z}$ doesn't have a closed-form solution and its numerical integration is infeasible
- This means that we cannot even evaluate the function $f(\boldsymbol{\theta})$ that we want to optimize (or its gradient $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta})$)! What can we do?

Recap: Normalizing Flows

$X \to Z$

- Note the difference to normalizing flows!
- Using reverse parametrization with a parametric function $g_{\boldsymbol{\theta}}(\boldsymbol{x})$ and base distribution p_1 we obtain

$$\max_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \max_{\boldsymbol{\theta}} \left[\log p_1\big(g_{\boldsymbol{\theta}}(\boldsymbol{x})\big) + \log \left| \det\left(\frac{\partial g_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right) \right| \right]$$

- This is tractable; we can even compute the gradient w.r.t. $\boldsymbol{\theta}$
  - easy using auto differentiation
  - the efficiency depends on structure of $g_{\boldsymbol{\theta}}(\boldsymbol{x})$

➢ Maximum Likelihood Estimation using NFs is tractable

Questions – VI1

1. Assume that we have an LVM, where $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ are tractable (i.e. we can compute them). Can it happen, that we can also compute $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ for this model, but cannot compute $p_{\boldsymbol{\theta}}(\boldsymbol{x})$? Why or why not?

2. Why is it always possible to compute $\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$ in a latent variable model, where $\boldsymbol{z}$ can take only finitely many values?

Roadmap

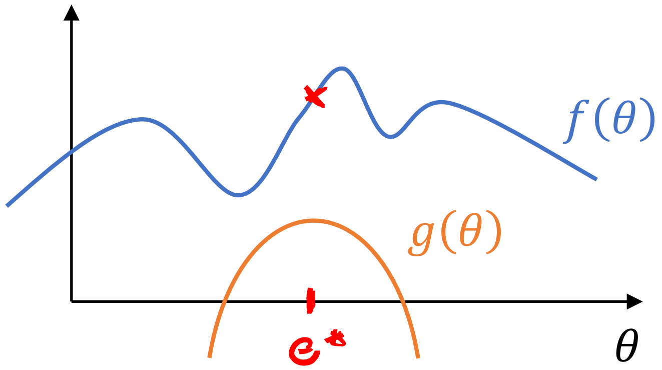
- Chapter: Deep Generative Models
  1. Introduction
  2. Normalizing Flows
  3. **Variational Inference**
     - Latent variable models
     - **Maximization using lower bounds**
     - Optimizing the ELBO
     - Variational Autoencoders
  4. Generative Adversarial Networks

Maximization using Lower Bounds

- We would like to solve a maximization problem

$$\max_{\theta} f(\theta)$$

  - Both f and ∇f are intractable (cannot be computed)



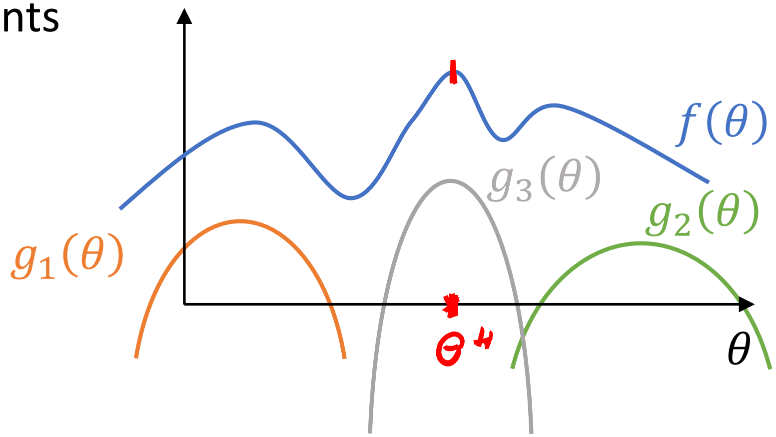
- Idea: Let's find some "nice" function $g(\theta)$ that is a lower bound on $f(\theta)$
  - That is, for all θ it holds that $f(\theta) \geq g(\theta)$

- Maximizing $g(\theta)$ would give us a lower bound on the solution of the original optimization problem

$$\max_{\theta} f(\theta) \geq \max_{\theta} g(\theta)$$

Multiple Lower Bounds

- Instead of using a single lower bound g, consider a collection $\mathcal{G}$ of lower bounds
  - For example, $\mathcal{G} = \{g_1, g_2, g_3\}$
  - Set $\mathcal{G}$ can contain uncountably many elements

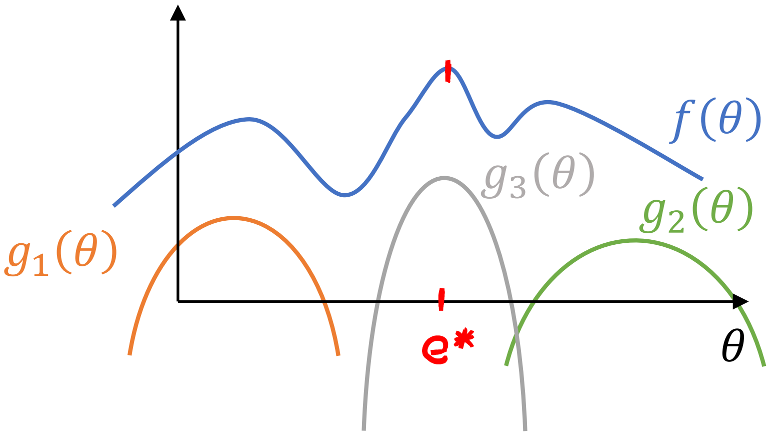


- Finding the best lower bound in $\mathcal{G}$ and maximizing it will get us even closer to the solution of the original problem

$$\max_{\theta} f(\theta) \geq \max_{g \in \mathcal{G}} \max_{\theta} g(\theta)$$

Maximization using Lower Bounds: Summary

- Algorithm: Approximately solving
$$\max_{\theta} f(\theta)$$
for some intractable function f



1. Construct a lower bound $g(\theta)$, such that
$$f(\theta) \geq g(\theta)$$
for all $g \in \mathcal{G}$ and for all θ

2. Solve the optimization problem
$$\max_{g \in \mathcal{G},\, \theta} g(\theta)$$

Lower Bound for the Marginal Log-likelihood

- How can we find a lower bound for $\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$?
- Let $q(\boldsymbol{z})$ be an arbitrary distribution over $\boldsymbol{z}$

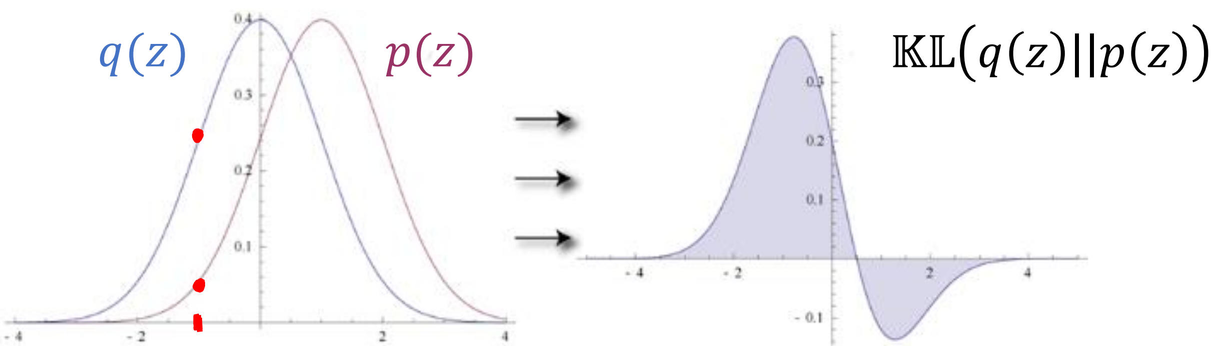
$$
\begin{aligned}
\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) &= \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x})] \\
&= \int q(\boldsymbol{z}) \log p_{\boldsymbol{\theta}}(\boldsymbol{x})\, d\boldsymbol{z} \\
&= \int q(\boldsymbol{z}) \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\, d\boldsymbol{z} \\
&= \int q(\boldsymbol{z}) \log \left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})} \cdot \frac{q(\boldsymbol{z})}{q(\boldsymbol{z})}\right) d\boldsymbol{z} \\
&= \int q(\boldsymbol{z}) \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z})}\, d\boldsymbol{z} + \int q(\boldsymbol{z}) \log \frac{q(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\, d\boldsymbol{z} \\
&= \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z})}\right] + \mathbb{KL}(q(\boldsymbol{z}) \,||\, p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}))
\end{aligned}
$$

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

Kullback–Leibler Divergence

- KL divergence from $q(\boldsymbol{z})$ to $p(\boldsymbol{z})$ is defined as

$$\mathbb{KL}\big(q(\boldsymbol{z})||p(\boldsymbol{z})\big) := \int q(\boldsymbol{z}) \log \frac{q(\boldsymbol{z})}{p(\boldsymbol{z})} d\boldsymbol{z}$$



Original Gaussian PDF's

KL Area to be Integrated

- Properties
  - Asymmetric, $\mathbb{KL}\big(q(\boldsymbol{z})||p(\boldsymbol{z})\big) \neq \mathbb{KL}\big(p(\boldsymbol{z})||\, q(\boldsymbol{z})\big)$ in general
  - Nonnegative, $\mathbb{KL}\big(q(\boldsymbol{z})||p(\boldsymbol{z})\big) \geq 0$
  - $\mathbb{KL}\big(q(\boldsymbol{z})||p(\boldsymbol{z})\big) = 0 \Leftrightarrow p = q$ almost everywhere

Evidence Lower BOund (ELBO)

- How can we find a lower bound for $\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$?
- Let $q(\boldsymbol{z})$ be an arbitrary distribution over $\boldsymbol{z}$

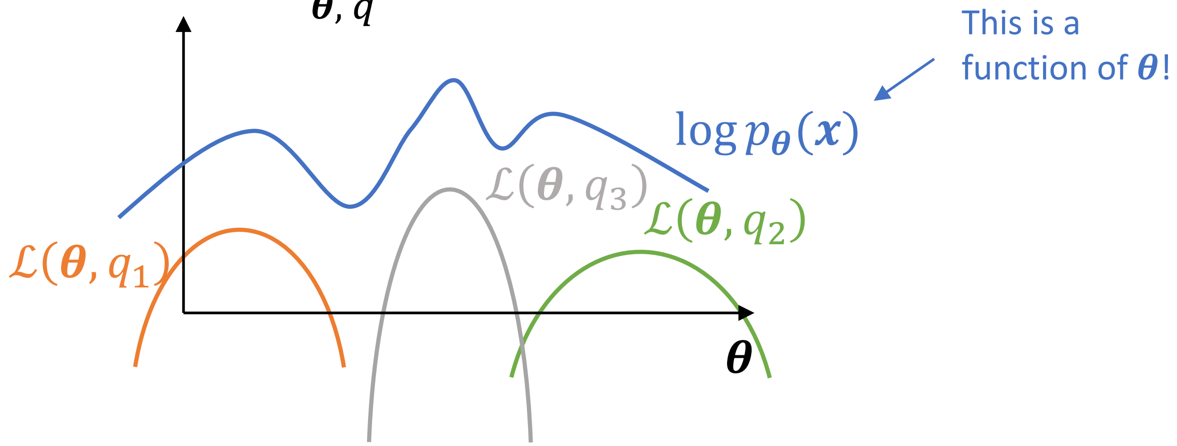
$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \underbrace{\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z})}\right]}_{\mathcal{L}(\boldsymbol{\theta},q)} + \underbrace{\mathbb{KL}\big(q(\boldsymbol{z})\,||\,p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\big)}_{\geq 0}$$

- Since KL divergence is nonnegative, $\mathcal{L}(\boldsymbol{\theta},q)$ is a lower bound on $\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$
- The expression $\log p_{\boldsymbol{\theta}}(x)$ is often called evidence, so we call $\mathcal{L}(\boldsymbol{\theta},q)$ Evidence Lower BOund (ELBO)
- The tightness of the bound depends on how close $q(\boldsymbol{z})$ is to the posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ in terms of KL divergence

Variational Inference

- We have derived a lower bound
$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq \mathbb{E}_z[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z})]$$
$$=: \mathcal{L}(\boldsymbol{\theta}, q)$$
  - Any distribution $q(\boldsymbol{z})$ defines a valid lower bound
  - Different choices of $q(\boldsymbol{z})$ lead to different lower bounds

- We need to find the parameters $\boldsymbol{\theta}$ and the distribution $q(\boldsymbol{z})$ that maximize the lower bound
$$\max_{\boldsymbol{\theta}, q} \mathcal{L}(\boldsymbol{\theta}, q)$$

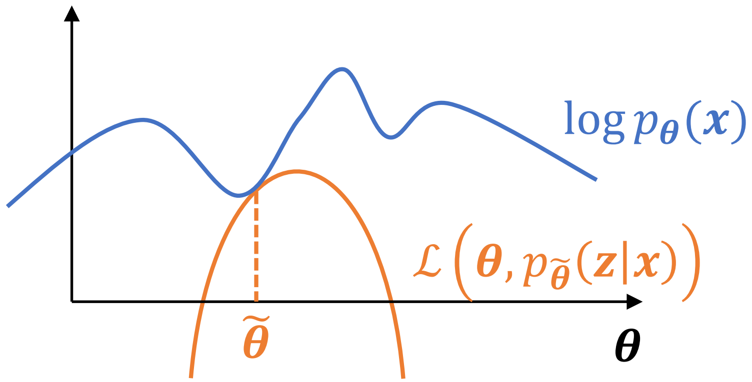


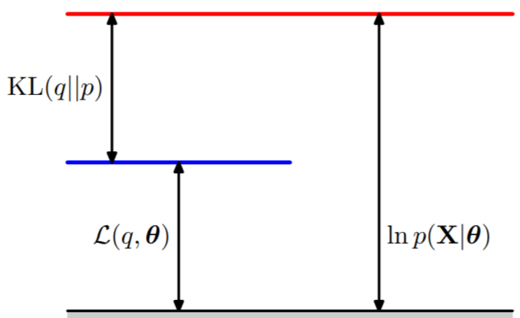
Alternative Interpretation of the ELBO

- We can equivalently rewrite the ELBO as following

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathcal{L}(\boldsymbol{\theta}, q) + \mathbb{KL}\big(q(\boldsymbol{z}) || p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\big)$$
$$\Rightarrow \mathcal{L}(\boldsymbol{\theta}, q) = -\mathbb{KL}\big(q(\boldsymbol{z}) || p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\big) + \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

- For any fixed $\boldsymbol{\theta}$, setting $q(\boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ will make ELBO exactly equal to $\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$ (i.e. our lower bound becomes tight at $\boldsymbol{\theta}$)



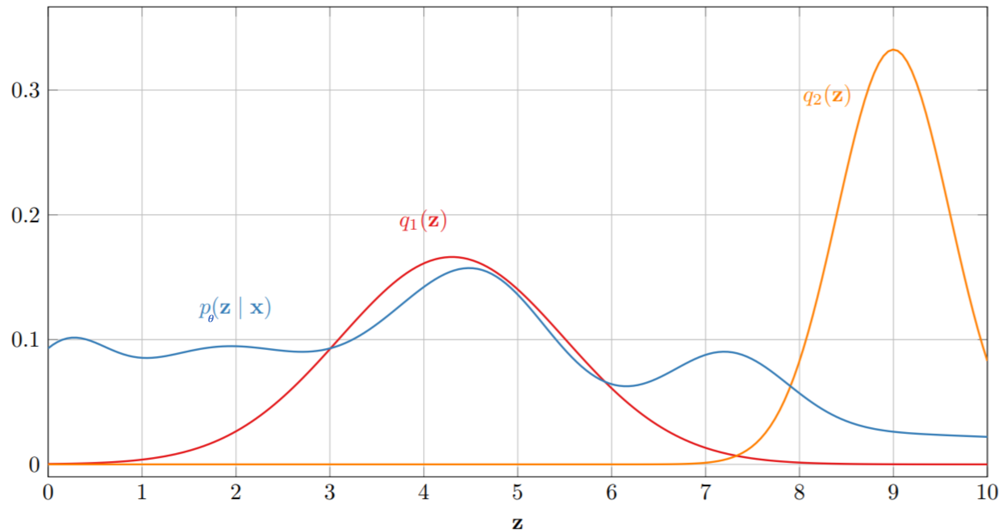


[2]

- In other words, for any fixed $\boldsymbol{\theta}$, maximizing the ELBO w.r.t. q is equivalent to making $q(\boldsymbol{z})$ as close as possible to $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ (in terms of KL divergence)

Intuitive Meaning of the ELBO

- The first distribution $q_1(\boldsymbol{z})$ is a good approximation to the true posterior
  - The KL divergence $\mathbb{KL}\big(q_1(\boldsymbol{z}) || p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\big)$ is low
  - The ELBO $\mathcal{L}(\boldsymbol{\theta}, q_1)$ is high
- The second distribution $q_2(\boldsymbol{z})$ is a bad approximation to the true posterior
  - The KL divergence $\mathbb{KL}\big(q_2(\boldsymbol{z}) || p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\big)$ is high
  - The ELBO $\mathcal{L}(\boldsymbol{\theta}, q_2)$ is low



EM Algorithm and Variational Inference

- Unfortunately, the true posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ is often also intractable, so we cannot just set $q(\boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$
- The models where we can compute $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ exactly are rather rare and remarkable.
  - Variational inference algorithm for such models even has a special name – Expectation Maximization (EM)

- The EM algorithm consists of two steps
  - E-step

$$\text{Set } q(\boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$$

  - M-step

$$\text{Set } \boldsymbol{\theta}^{\text{new}} = \operatorname{argmax}_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z} \sim q(\boldsymbol{z})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})]$$

- We can verify that this procedure indeed maximizes the ELBO

EM Algorithm and Variational Inference

- The ELBO is defined as

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}, q) &= \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z})] \\ &= -\mathbb{KL}\big(q(\boldsymbol{z}) || p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\big) + \log p_{\boldsymbol{\theta}}(\boldsymbol{x})\end{aligned}$$

- E-step
  - Set $q(\boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) = \operatorname*{argmin}_q \mathbb{KL}\big(q(\boldsymbol{z}) || p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\big) = \operatorname{argmax}_{\boldsymbol{q}} \mathcal{L}(\boldsymbol{\theta}, q)$
  - Making $q(\boldsymbol{z})$ equal to $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ minimizes the KL divergence from $q(\boldsymbol{z})$ to $p(\boldsymbol{z}|\boldsymbol{x})$, thus maximizing the ELBO w.r.t. q

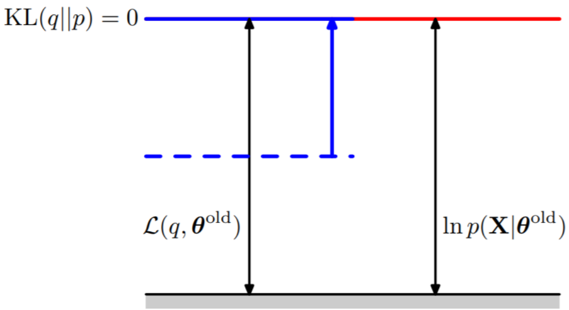
- M-step
  - Set $\boldsymbol{\theta}^{\text{new}} = \operatorname{argmax}_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})] = \operatorname{argmax}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, q)$
  - This maximizes the ELBO w.r.t. $\boldsymbol{\theta}$ while keeping q fixed

- The EM algorithm is just doing alternating optimization of the ELBO in a model, where $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ can be computed exactly!

EM Algorithm and Variational Inference

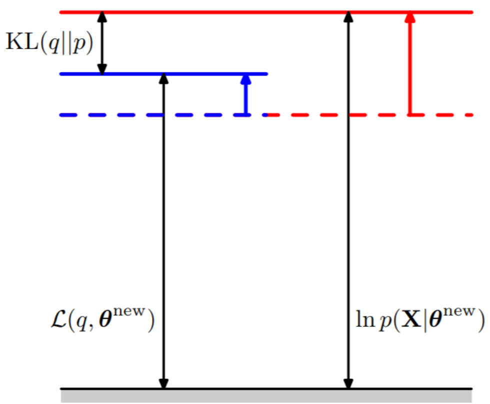
- E-step

$$q(\boldsymbol{z}) = p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$$
$$= \underset{q}{\operatorname{argmin}}\, \mathbb{KL}\big(q(\boldsymbol{z})||p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\big)$$
$$= \operatorname{argmax}_{\boldsymbol{q}}\, \mathcal{L}(\boldsymbol{\theta}, q)$$



- M-step

$$\boldsymbol{\theta}^{\text{new}} = \operatorname{argmax}_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})]$$
$$= \operatorname{argmax}_{\boldsymbol{\theta}}\, \mathcal{L}(\boldsymbol{\theta}, q)$$



[2]

Questions – VI2

1. Slide 57: Why is it necessary for all functions $g \in \mathcal{G}$ to be lower bounds on f? What happens if some functions in $\mathcal{G}$ are not lower bounds?

2. Slide 57: Can we use a similar approach if we want to approximately <u>minimize</u> some intractable function f? What changes need to be done in this case?

3. Assume that $p_\theta(z|x)$ is a distribution on $[0, \infty)$ (e.g. exponential distribution), and our variational distribution $q(z)$ is a distribution on all of $\mathbb{R}$ (e.g. normal distribution).
   - What happens to the ELBO in this case?
   - Why is the optimization problem of maximizing the ELBO ill-defined?
   - How can we fix this problem?

Roadmap

- Chapter: Deep Generative Models
  1. Introduction
  2. Normalizing Flows
  3. **Variational Inference**
     – Latent variable models
     – Maximization using lower bounds
     – **Optimizing the ELBO**
     – Variational Autoencoders
  4. Generative Adversarial Networks

Optimizing the ELBO

- How do we actually solve this optimization problem?

$$\max_{\boldsymbol{\theta}, q} \mathcal{L}(\boldsymbol{\theta}, q)$$

- $\boldsymbol{\theta} \in \mathbb{R}^M$ is just a vector, so we know how maximize with respect to it

- However, $q(\boldsymbol{z})$ is a probability distribution. This leads to two questions:
  1. What is the domain that we are optimizing over?
  2. How can we optimize w.r.t. a probability distribution?

Parametric Family of Distributions

- We pick a set of candidate tractable parametric distributions $\mathcal{Q}$
$$\max_{\boldsymbol{\theta}\in\mathbb{R}^M,\, q\in\mathcal{Q}} \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z}) - \log q(\boldsymbol{z})]$$
- Tractable: We can draw samples from q and compute the density $q(\boldsymbol{z})$
- Parametric: every distribution in $\mathcal{Q}$ is specified by its parameter vector $\boldsymbol{\phi} \in \mathbb{R}^K$
  - $\mathcal{Q} = \left\{q_{\boldsymbol{\phi}}(\boldsymbol{z}) \text{ for } \boldsymbol{\phi} \in \mathbb{R}^K\right\}$
  - We may also have constraints on $\boldsymbol{\phi}$ (e.g. nonnegativity), in that case $\boldsymbol{\phi} \in \mathcal{F} \subseteq \mathbb{R}^K$

Parametric Family of Distributions

- Some examples:
  - "$\mathcal{Q}$ is the set of all 2D normal distributions with identity covariance" or in mathematical notation $\mathcal{Q} = \{\mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}, \boldsymbol{I}_2) \text{ for } \boldsymbol{\mu} \in \mathbb{R}^2\}$. Here, $\boldsymbol{\phi} = \boldsymbol{\mu}$
  - "$\mathcal{Q}$ is the set of all 1D exponential distributions", or $\mathcal{Q} = \{\text{Expo}(\lambda) \text{ for } \lambda \in \mathbb{R}_{>0}\}$. Here, $\phi = \lambda$
  - $\mathcal{Q}$ is the set of all distributions that can be modelled via a Normalizing Flow with forward parametrization based on f_ϕ $\quad f_\phi(z) = x \quad z \sim \mathcal{N}(0,1)$

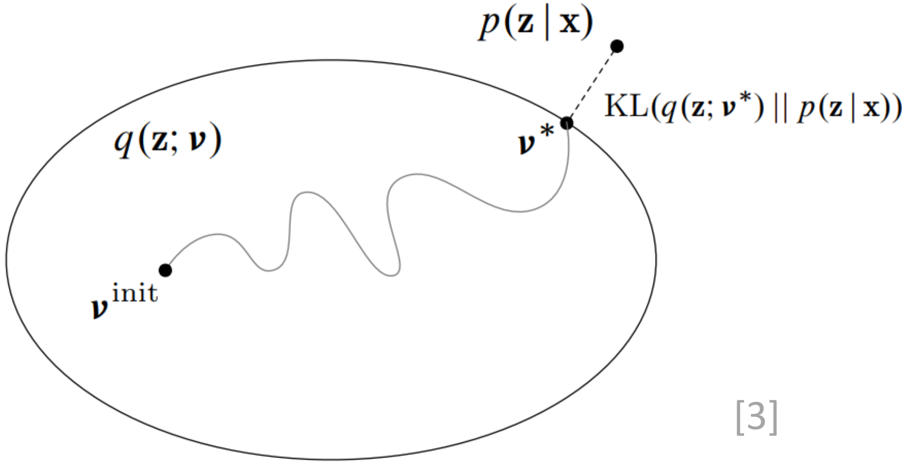
- Finding the "best" distribution $q \in \mathcal{Q} \Leftrightarrow$ finding the "best" parameters $\boldsymbol{\phi} \in \mathbb{R}^K$

- Now the variables in our optimization problem are just vectors

$$\max_{\boldsymbol{\theta} \in \mathbb{R}^M,\, \boldsymbol{\phi} \in \mathbb{R}^K} \mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z})\right]$$

Optimizing in the Space of Distributions

- Remember that for a fixed $\boldsymbol{\theta}$, maximizing the ELBO w.r.t. q is equivalent to minimizing $\mathbb{KL}\left(q_{\boldsymbol{\phi}}(\boldsymbol{z}) || p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})\right)$
- The true posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ is usually intractable – it's not contained in our tractable parametric family $\mathcal{Q} = \left\{q_{\boldsymbol{\phi}}(\boldsymbol{z}) \text{ for } \boldsymbol{\phi} \in \mathbb{R}^K\right\}$
- Optimizing over $\boldsymbol{\phi}$ leads to finding the distribution $q \in \mathcal{Q}$ that is the closest to the true posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ in terms of KL divergence
  - The word "variational" represents the fact that we are optimizing over distributions (functions)
  - The word "inference" – "we are doing approximate inference of $\boldsymbol{z}$ given $\boldsymbol{x}$"



[3]

Reformulated Optimization Problem

- We want to maximize the ELBO w.r.t. $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$
$$\max_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z})\right] =: \max_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi})$$
  - $\boldsymbol{\theta} \in \mathbb{R}^M$ are the parameters of our probabilistic model
  - $\boldsymbol{\phi} \in \mathbb{R}^K$ are the parameters of the variational distribution q

- This seems almost like a regular optimization problem, except that the objective function is a bit weird – it contains expectations (i.e. integrals)

- We can use standard tools from continuous optimization such as gradient ascent

- For this we simply need to compute $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi})$ and $\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi})$

Gradients of the ELBO

- The expectation in the ELBO is just an integral

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) &= \mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z})\right] \\ &= \int \left(\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z})\right) q_{\boldsymbol{\phi}}(\boldsymbol{z}) d\boldsymbol{z}\end{aligned}$$

- For some simple models this integral can be computed analytically

- In this case, we can simply compute $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$ and $\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$ by hand or using autodifferentiation libraries (e.g. PyTorch or TensorFlow)

- Given the gradients, just optimize the ELBO like any other function

- What if the integral (i.e. expectation) cannot be computed analytically?

$f(\theta, z)$

Approximating $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$

- Let's assume that $\boldsymbol{\phi}$ is known and fixed, and we only want to find $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$
- This is an instance of a more general problem

$$\mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}[f_{\boldsymbol{\theta}}(\boldsymbol{z})] = \int q_{\boldsymbol{\phi}}(\boldsymbol{z}) f_{\boldsymbol{\theta}}(\boldsymbol{z}) d\boldsymbol{z}$$

  – In our case $f_{\boldsymbol{\theta}}(\boldsymbol{z}) = \log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})$

- We can approximate the integral using Monte Carlo

$$\mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}[f_{\boldsymbol{\theta}}(\boldsymbol{z})] \approx \frac{1}{S} \sum_{i=1}^{S} f_{\boldsymbol{\theta}}(\boldsymbol{z}_i) \quad \text{where } \boldsymbol{z}_i \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}) \text{ for } i = 1, \dots, S$$

- Approximating the gradient is just as easy

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}[f_{\boldsymbol{\theta}}(\boldsymbol{z})] = \nabla_{\boldsymbol{\theta}} \int q_{\boldsymbol{\phi}}(\boldsymbol{z}) f_{\boldsymbol{\theta}}(\boldsymbol{z}) d\boldsymbol{z} = \int q_{\boldsymbol{\phi}}(\boldsymbol{z}) \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{z}) d\boldsymbol{z}$$

$$= \mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}[\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{z})] \approx \frac{1}{S} \sum_{i=1}^{S} \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{z}_i)$$

$h_\phi(z) = z^2$

Approximating $\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$

$\mathbb{E}_{z \sim \mathcal{N}(\mu, ?)}[h_\phi(z)]$

- Now, assume that $\boldsymbol{\theta}$ is known and we want to compute $\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$
- Again, let's look at a more general formulation

$$\mathbb{E}_{\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z})}[h_{\boldsymbol{\phi}}(\mathbf{z})] = \int q_{\boldsymbol{\phi}}(\mathbf{z}) h_{\boldsymbol{\phi}}(\mathbf{z}) d\mathbf{z}$$

  – In our case $h_{\boldsymbol{\phi}}(\mathbf{z}) = \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z})$

- In this case, we cannot just "push" the gradient inside the integral

$$\nabla_{\boldsymbol{\phi}} \mathbb{E}_{\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z})}[h_{\boldsymbol{\phi}}(\mathbf{z})] = \nabla_{\boldsymbol{\phi}} \int q_{\boldsymbol{\phi}}(\mathbf{z}) h_{\boldsymbol{\phi}}(\mathbf{z}) d\mathbf{z}$$

$$\neq \int q_{\boldsymbol{\phi}}(\mathbf{z}) \nabla_{\boldsymbol{\phi}} h_{\boldsymbol{\phi}}(\mathbf{z}) d\mathbf{z} = \mathbb{E}_{\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z})}\left[\nabla_{\boldsymbol{\phi}} h_{\boldsymbol{\phi}}(\mathbf{z})\right]$$

- The gradient $\nabla_{\boldsymbol{\phi}}$ should also somehow act on $q_{\boldsymbol{\phi}}(\mathbf{z})$!
  – Think about what happens if $h_{\boldsymbol{\phi}}(\mathbf{z}) = h(\mathbf{z})$, i.e. h doesn't depend on $\boldsymbol{\phi}$
- How can we approximate the gradient of the expectation is this case?

Reparametrization Trick

- Idea: Sampling from many distributions $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ can be represented as a deterministic transformation $T(\boldsymbol{\epsilon}, \boldsymbol{\phi})$ of some base distribution $b(\boldsymbol{\epsilon})$
- For example, let $q_{\boldsymbol{\phi}}(z) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}, \boldsymbol{R}\boldsymbol{R}^T)$ be a multivariate normal distribution
- Sampling from $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ is equivalent to
  1. Drawing a sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
  2. Obtaining $\boldsymbol{z} = T(\boldsymbol{\epsilon}, \boldsymbol{\phi} = \{\boldsymbol{\mu}, \boldsymbol{R}\}) = \boldsymbol{R}\boldsymbol{\epsilon} + \boldsymbol{\mu}$

See again: Normalizing Flows!

- Important: The distribution $b(\boldsymbol{\epsilon})$ does not depend on $\boldsymbol{\phi}$
- This trick will allow us to compute gradients w.r.t. $\boldsymbol{\phi}$

Reparametrization Trick in Action

- Using the reparametrization trick, we can rewrite our expectation as

$$\begin{aligned}\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[h_{\boldsymbol{\phi}}(\boldsymbol{z})\right] &= \int q_{\boldsymbol{\phi}}(\boldsymbol{z})h_{\boldsymbol{\phi}}(\boldsymbol{z})d\boldsymbol{z} \\ &= \int b(\boldsymbol{\epsilon})h_{\boldsymbol{\phi}}(T(\boldsymbol{\epsilon},\boldsymbol{\phi}))d\boldsymbol{\epsilon} \\ &= \mathbb{E}_{\boldsymbol{\epsilon}\sim b(\boldsymbol{\epsilon})}\left[h_{\boldsymbol{\phi}}\left(T(\boldsymbol{\epsilon},\boldsymbol{\phi})\right)\right]\end{aligned}$$

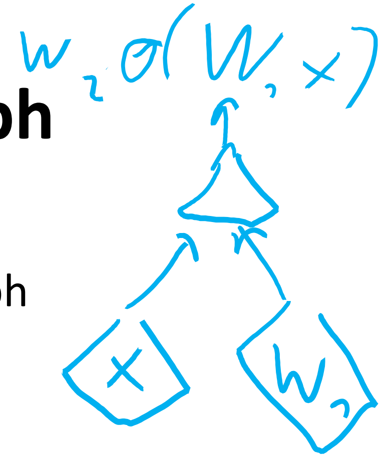
- This is exactly the situation that we had with $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi})$!

$$\nabla_{\boldsymbol{\phi}}\mathbb{E}_{\boldsymbol{\epsilon}\sim b(\boldsymbol{\epsilon})}\left[h_{\boldsymbol{\phi}}\left(T(\boldsymbol{\epsilon},\boldsymbol{\phi})\right)\right] = \nabla_{\boldsymbol{\phi}}\int b(\boldsymbol{\epsilon})h_{\boldsymbol{\phi}}\left(T(\boldsymbol{\epsilon},\boldsymbol{\phi})\right)d\boldsymbol{\epsilon} = \int b(\boldsymbol{\epsilon})\nabla_{\boldsymbol{\phi}}h_{\boldsymbol{\phi}}\left(T(\boldsymbol{\epsilon},\boldsymbol{\phi})\right)d\boldsymbol{\epsilon}$$

$$= \mathbb{E}_{\boldsymbol{\epsilon}\sim b(\boldsymbol{\epsilon})}\left[\nabla_{\boldsymbol{\phi}}h_{\boldsymbol{\phi}}\left(T(\boldsymbol{\epsilon},\boldsymbol{\phi})\right)\right] \approx \frac{1}{S}\sum_{i=1}^{S}\nabla_{\boldsymbol{\phi}}h_{\boldsymbol{\phi}}\left(T(\boldsymbol{\epsilon}_i,\boldsymbol{\phi})\right) \text{ where } \boldsymbol{\epsilon}_i\sim b(\boldsymbol{\epsilon}) \text{ for } i = 1,\ldots,S$$

- This is possible because $b(\boldsymbol{\epsilon})$ doesn't depend on $\boldsymbol{\phi}$

Reparametrization Trick & Computation Graph



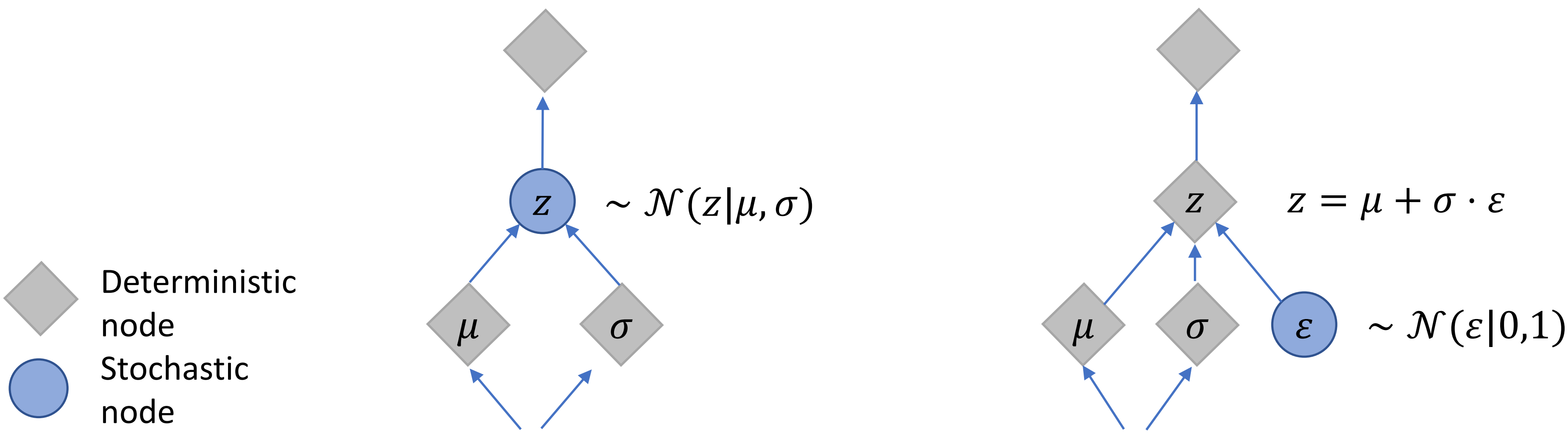
- Assume you have the following operation in your computation graph

$$f(\boldsymbol{z}, \dots) \text{ where } \boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})$$

- To allow backpropagation/gradient computation, reformulate to

$$f(T(\boldsymbol{\epsilon}, \boldsymbol{\phi}), \dots) \text{ where } \boldsymbol{\epsilon} \sim b(\boldsymbol{\epsilon})$$

- Important:
  - The distribution $b(\boldsymbol{\epsilon})$ does not depend on $\boldsymbol{\phi}$ (or any other variable we are optimizing over)
  - It has no predecessors in the computation graph



Putting Everything Together

1. We define a latent variable generative model for our data $\boldsymbol{x}$

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

2. We are interested in maximum likelihood estimation of model parameters $\boldsymbol{\theta}$

$$\max_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x})$$

3. We can obtain a lower bound on $\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$ using some distribution $q(\boldsymbol{z})$

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq \mathcal{L}(\boldsymbol{\theta}, q) \coloneqq \mathbb{E}_{\boldsymbol{z} \sim q(\boldsymbol{z})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z})]$$

4. Our original optimization problem can be approximately solved as

$$\max_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq \max_{\boldsymbol{\theta}, q} \mathcal{L}(\boldsymbol{\theta}, q)$$

Putting Everything Together

5. We pick a parametric family of variational distributions $\mathcal{Q}$
$$\mathcal{Q} = \{q_{\boldsymbol{\phi}}(\boldsymbol{z}) \ \text{ for } \boldsymbol{\phi} \in \mathbb{R}^K\}$$

6. Our optimization problem now becomes
$$\max_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi}) \coloneqq \max_{\boldsymbol{\theta},\boldsymbol{\phi}} \ \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{z})\right]$$

7. We obtain the gradients $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi})$ and $\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi})$ using Monte Carlo (with the reparametrization trick)

8. We find $\boldsymbol{\theta}^\star, \boldsymbol{\phi}^\star$ by maximizing our objective function using gradient ascent

Dealing with the Entire Dataset

$q(z_i | x_i, X)$

- There is one important detail that we haven't covered so far
- Usually, we learn our models using a dataset $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ that contains multiple samples
- Our actual optimization problem is

$$\max_\theta \frac{1}{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{X}) = \max_\theta \frac{1}{N} \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$$

- In order to lower bound $\log p_{\boldsymbol{\theta}}(\boldsymbol{X})$, we need to consider the distribution $q(\boldsymbol{Z})$ over all the latent variables $\boldsymbol{Z} = \{\boldsymbol{z}_i\}_{i=1}^N$ for all the instances i

- In this case, the ELBO is

$$\frac{1}{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{X}) \geq \frac{1}{N} \mathbb{E}_{\boldsymbol{Z} \sim q(\boldsymbol{Z})} [\log p_{\boldsymbol{\theta}}(\boldsymbol{X}, \boldsymbol{Z}) - \log q(\boldsymbol{Z})]$$

$$= \frac{1}{N} \mathbb{E}_{\boldsymbol{Z} \sim q(\boldsymbol{Z})} \left[\sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{z}_i) - \log q(\boldsymbol{Z}) \right]$$

Mean Field Assumption

x_1 | $q_1^* = \mathcal{N}(3, 1)$ ϕ_1

x_2 | $q_2^* = \mathcal{N}(4, 2)$ ϕ_2

- Note, that in general $q(\boldsymbol{Z})$ allows to have dependencies between the latent variables $\boldsymbol{z}_i$ for different data points i
- In practice we often make a simplifying assumption that $q(\boldsymbol{Z})$ factorizes

$$q(\boldsymbol{Z}) = \prod_{i=1}^{N} q_i(\boldsymbol{z}_i)$$

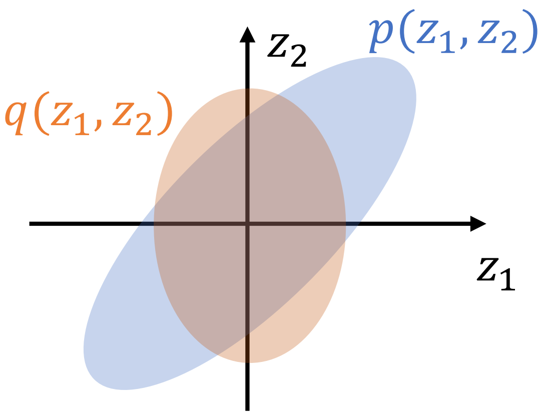
- This assumption is often called "mean field" (for historical reasons that are not particularly interesting, unless you are a physicist)
- The two main advantages of such assumption are
  - It's easier to model the distribution $q(\boldsymbol{z}_i)$ over $\boldsymbol{z}_i \in \mathbb{R}^L$, than $q(\boldsymbol{Z})$ over $\boldsymbol{Z} \in \mathbb{R}^{N \times L}$
  - The ELBO simplifies

$q_{\phi_i}(z_i)$

$$\frac{1}{N}\mathbb{E}_{\boldsymbol{Z}\sim q(\boldsymbol{Z})}\left[\sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{z}_i) - \log q(\boldsymbol{Z})\right] = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{z}_i\sim q_i(\boldsymbol{z}_i)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{z}_i) - \log q_i(\boldsymbol{z}_i)]$$

Implications of the Mean Field Assumption

- What does this mean to have a distribution that factorizes?
  - We cannot capture the dependencies / correlations between dimensions
  - Our approximate posterior is less expressive (we can only represent a subset of distributions), but optimization and inference become easier



- Example in 2D
  - The blue distribution $p(z_1, z_2)$ cannot be factorized
  - The orange distribution can be written as $q(z_1)q(z_2)$

- If our data is i.i.d., this assumption is a often a pretty good approximation
  - Consider, e.g. a collection of images $\boldsymbol{x}_i$: latent features $\boldsymbol{z}_i \in \mathbb{R}^L$ of image i (lighting, scene composition) do not depend on the latent features of image j

- If the true posterior is highly correlated, the approximation can be poor
  - Consider a social network: if we know about the latent features $\boldsymbol{z}_i$ of node i (e.g. this person is a student), this gives us some information about its neighbors – it's likely that they are students too

$\varepsilon_1, \varepsilon_2, \varepsilon_3$

Mean Field Assumption for Parametric Distributions

θ, ϕ_1, ϕ_2

$\varepsilon_1', \varepsilon_2'$

- In a parametric model, the mean field assumption implies

$$q_{\boldsymbol{\phi}}(\boldsymbol{Z}) = \prod_{i}^{N} q_{\boldsymbol{\phi}_i}(\boldsymbol{z}_i)$$

- This means, we have to learn a parameter vector $\boldsymbol{\phi}_i \in \mathbb{R}^K$ for each instance i in our dataset ($N \times K$ parameters in total)
- The ELBO is

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_i(\boldsymbol{\theta}, \boldsymbol{\phi}_i) \coloneqq \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\boldsymbol{z}_i \sim q_{\boldsymbol{\phi}_i}(\boldsymbol{z}_i)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{z}_i) - \log q_{\boldsymbol{\phi}_i}(\boldsymbol{z}_i)]$$

- This simplification of ELBO allows us to approximate $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$ using a mini-batch of data points with indices $\mathcal{B} \subseteq \{1, \dots, N\} \Rightarrow$ even more efficient training

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \frac{1}{N}\sum_{i=1}^{N} \nabla_{\boldsymbol{\theta}}\mathcal{L}_i(\boldsymbol{\theta}, \boldsymbol{\phi}_i) \approx \frac{1}{|\mathcal{B}|}\sum_{i \in \mathcal{B}} \nabla_{\boldsymbol{\theta}}\mathcal{L}_i(\boldsymbol{\theta}, \boldsymbol{\phi}_i)$$

Questions – VI3

1. Which of the following conditions have to be satisfied by a distribution $q(\boldsymbol{z})$, such that it's possible to use it in variational inference (as described in our recipe on slides 80-81)
   a) We can compute the expectated value of $\boldsymbol{z}$ in closed form
   b) We can compute the entropy of $q(\boldsymbol{z})$ in closed form
   c) We can draw samples from $q(\boldsymbol{z})$ with reparametrization
   d) We can compute the density $\log q(\boldsymbol{z})$ for an arbitrary $\boldsymbol{z}$
   e) We can compute $\log q(\boldsymbol{z})$ for a sample $\boldsymbol{z}$ drawn from $q(\boldsymbol{z})$
   f) The distribution can be factorized as $q(\boldsymbol{z}) = \prod_i q_i(\boldsymbol{z}_i)$

2. Think of a probabilistic model with two latent variables $z_1, z_2 \in \mathbb{R}$ and an observed variable $x \in \mathbb{R}$ (i.e. write down $p_{\boldsymbol{\theta}}(x|z_1, z_2)$ and $p_{\boldsymbol{\theta}}(z_1, z_2)$), where the posterior can be factorized as $p_{\boldsymbol{\theta}}(z_1, z_2|x) = p_{\boldsymbol{\theta}}(z_1|x)p_{\boldsymbol{\theta}}(z_2|x)$.

3. Same as question 3, but now the posterior cannot be factorized.

Reading Materials

- Unfortunately, we are not aware of a good up-to-date reference that thoroughly presents the modern view on variational inference for learning generative models. Two slightly outdated, but still decent resources are

1. C. Bishop "Pattern Recognition and Machine Learning" – Section 9.4
2. D. Blei et al., "Variational Inference: A Review for Statisticians", https://arxiv.org/abs/1601.00670

References for Figures

1. https://fallfordata.com/soft-clustering-with-gaussian-mixture-models-gmm/
2. C. Bishop, “Pattern Recognition and Machine Learning”, 2006
3. D. Blei et al., https://media.nips.cc/Conferences/2016/Slides/6199-Slides.pdf