Deep Generative Models

Lecture 3

Roman Isachenko

Moscow Institute of Physics and Technology
Yandex School of Data Analysis

2024, Autumn

Recap of previous lecture

Jacobian matrix

Let $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^m$ be a differentiable function.

$$\mathbf{z} = \mathbf{f}(\mathbf{x}), \quad \mathbf{J} = \frac{\partial \mathbf{z}}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial z_1}{\partial x_1} & \dots & \frac{\partial z_1}{\partial x_m} \\ \dots & \dots & \dots \\ \frac{\partial z_m}{\partial x_1} & \dots & \frac{\partial z_m}{\partial x_m} \end{pmatrix} \in \mathbb{R}^{m \times m}$$

Change of variable theorem (CoV)

Let $\mathbf{x}$ be a random variable with density function $p(\mathbf{x})$ and $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^m$ is a differentiable, invertible function. If $\mathbf{z} = \mathbf{f}(\mathbf{x})$, $\mathbf{x} = \mathbf{f}^{-1}(\mathbf{z}) = \mathbf{g}(\mathbf{z})$, then

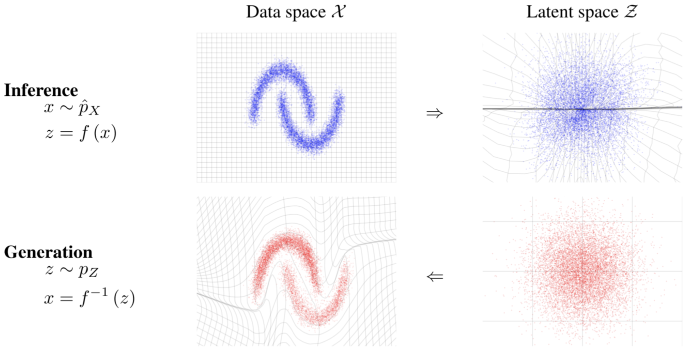
$$p(\mathbf{x}) = p(\mathbf{z})|\det(\mathbf{J}_\mathbf{f})| = p(\mathbf{z}) \left|\det\left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}}\right)\right| = p(\mathbf{f}(\mathbf{x})) \left|\det\left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right)\right|$$

$$p(\mathbf{z}) = p(\mathbf{x})|\det(\mathbf{J}_\mathbf{g})| = p(\mathbf{x}) \left|\det\left(\frac{\partial \mathbf{x}}{\partial \mathbf{z}}\right)\right| = p(\mathbf{g}(\mathbf{z})) \left|\det\left(\frac{\partial \mathbf{g}(\mathbf{z})}{\partial \mathbf{z}}\right)\right|.$$

Recap of previous lecture

Definition

Normalizing flow is a *differentiable, invertible* mapping from data $\mathbf{x}$ to the noise $\mathbf{z}$.



Log likelihood

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(\mathbf{f}_K \circ \dots \circ \mathbf{f}_1(\mathbf{x})) + \sum_{k=1}^{K} \log |\det(\mathbf{J}_{\mathbf{f}_k})|$$

Dinh L., Sohl-Dickstein J., Bengio S. Density estimation using Real NVP, 2016

Recap of previous lecture

Flow log-likelihood

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})) + \log |\det(\mathbf{J}_{\mathbf{f}})|$$

The main challenge is a determinant of the Jacobian.

Linear flows

$$\mathbf{z} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{W}\mathbf{x}, \quad \mathbf{W} \in \mathbb{R}^{m \times m}, \quad \boldsymbol{\theta} = \mathbf{W}, \quad \mathbf{J}_{\mathbf{f}} = \mathbf{W}^T$$

- LU-decomposition
$$\mathbf{W} = \mathbf{P}\mathbf{L}\mathbf{U}.$$
- QR-decomposition
$$\mathbf{W} = \mathbf{Q}\mathbf{R}.$$

Decomposition should be done only once in the beggining. Next, we fit decomposed matrices ($\mathbf{P}/\mathbf{L}/\mathbf{U}$ or $\mathbf{Q}/\mathbf{R}$).

Kingma D. P., Dhariwal P. Glow: Generative Flow with Invertible 1x1 Convolutions, 2018

Hoogeboom E., et al. Emerging convolutions for generative normalizing flows, 2019

Recap of previous lecture

Consider an autoregressive model

$$p(\mathbf{x}|\boldsymbol{\theta}) = \prod_{j=1}^{m} p(x_j|\mathbf{x}_{1:j-1}, \boldsymbol{\theta}), \quad p(x_j|\mathbf{x}_{1:j-1}, \boldsymbol{\theta}) = \mathcal{N}\left(\mu_{j,\boldsymbol{\theta}}(\mathbf{x}_{1:j-1}), \sigma^2_{j,\boldsymbol{\theta}}(\mathbf{x}_{1:j-1})\right).$$

Gaussian autoregressive NF

$$\mathbf{x} = \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{z}) \quad \Rightarrow \quad x_j = \sigma_{j,\boldsymbol{\theta}}(\mathbf{x}_{1:j-1}) \cdot z_j + \mu_{j,\boldsymbol{\theta}}(\mathbf{x}_{1:j-1}).$$

$$\mathbf{z} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}) \quad \Rightarrow \quad z_j = (x_j - \mu_{j,\boldsymbol{\theta}}(\mathbf{x}_{1:j-1})) \cdot \frac{1}{\sigma_{j,\boldsymbol{\theta}}(\mathbf{x}_{1:j-1})}.$$

- We have an **invertible** and **differentiable** transformation from $p(\mathbf{z})$ to $p(\mathbf{x}|\boldsymbol{\theta})$.
- Jacobian of such transformation is triangular!

Generation function $\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{z})$ is **sequential**.
Inference function $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$ is **not sequential**.

Papamakarios G., Pavlakou T., Murray I. Masked Autoregressive Flow for Density Estimation, 2017

Recap of previous lecture

Let split $\mathbf{x}$ and $\mathbf{z}$ in two parts:

$$\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2] = [\mathbf{x}_{1:d}, \mathbf{x}_{d+1:m}]; \quad \mathbf{z} = [\mathbf{z}_1, \mathbf{z}_2] = [\mathbf{z}_{1:d}, \mathbf{z}_{d+1:m}].$$

Coupling layer

$$\begin{cases} \mathbf{x}_1 = \mathbf{z}_1; \\ \mathbf{x}_2 = \mathbf{z}_2 \odot \boldsymbol{\sigma}_{\boldsymbol{\theta}}(\mathbf{z}_1) + \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}_1). \end{cases} \quad \begin{cases} \mathbf{z}_1 = \mathbf{x}_1; \\ \mathbf{z}_2 = (\mathbf{x}_2 - \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_1)) \odot \frac{1}{\boldsymbol{\sigma}_{\boldsymbol{\theta}}(\mathbf{x}_1)}. \end{cases}$$

Estimating the density takes 1 pass, sampling takes 1 pass!

Jacobian

$$\det \left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right) = \det \begin{pmatrix} \mathbf{I}_d & 0_{d \times m-d} \\ \frac{\partial \mathbf{z}_2}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{z}_2}{\partial \mathbf{x}_2} \end{pmatrix} = \prod_{j=1}^{m-d} \frac{1}{\sigma_{j,\boldsymbol{\theta}}(\mathbf{x}_1)}.$$

Coupling layer is a special case of autoregressive NF.

Dinh L., Sohl-Dickstein J., Bengio S. Density estimation using Real NVP, 2016

Outline

1. Forward and Reverse KL for NF
2. Latent variable models (LVM)
3. Variational lower bound (ELBO)
4. EM-algorithm
5. Amortized inference

Outline

Forward KL vs Reverse KL

Forward KL $\equiv$ MLE

$$
\begin{aligned}
KL(\pi || p) &= \int \pi(\mathbf{x}) \log \frac{\pi(\mathbf{x})}{p(\mathbf{x}|\boldsymbol{\theta})} d\mathbf{x} \\
&= -\mathbb{E}_{\pi(\mathbf{x})} \log p(\mathbf{x}|\boldsymbol{\theta}) + \text{const} \to \min_{\boldsymbol{\theta}}
\end{aligned}
$$

Forward KL for NF model

$$
\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})) + \log |\det(\mathbf{J}_{\mathbf{f}})|
$$

$$
KL(\pi || p) = -\mathbb{E}_{\pi(\mathbf{x})} \left[\log p(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})) + \log |\det(\mathbf{J}_{\mathbf{f}})|\right] + \text{const}
$$

- We need to be able to compute $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$ and its Jacobian.
- We need to be able to compute the density $p(\mathbf{z})$.
- We don't need to think about computing the function $\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{z}) = \mathbf{f}_{\boldsymbol{\theta}}^{-1}(\mathbf{z})$ until we want to sample from the NF.

Forward KL vs Reverse KL

Reverse KL

$$
\begin{aligned}
KL(p||\pi) &= \int p(\mathbf{x}|\boldsymbol{\theta}) \log \frac{p(\mathbf{x}|\boldsymbol{\theta})}{\pi(\mathbf{x})} d\mathbf{x} \\
&= \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\theta})} \left[\log p(\mathbf{x}|\boldsymbol{\theta}) - \log \pi(\mathbf{x})\right] \rightarrow \min_{\boldsymbol{\theta}}
\end{aligned}
$$

Reverse KL for NF model (LOTUS trick)

$$
\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(\mathbf{z}) + \log |\det(\mathbf{J_f})| = \log p(\mathbf{z}) - \log |\det(\mathbf{J_g})|
$$

$$
KL(p||\pi) = \mathbb{E}_{p(\mathbf{z})} \left[\log p(\mathbf{z}) - \log |\det(\mathbf{J_g})| - \log \pi(\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{z}))\right]
$$

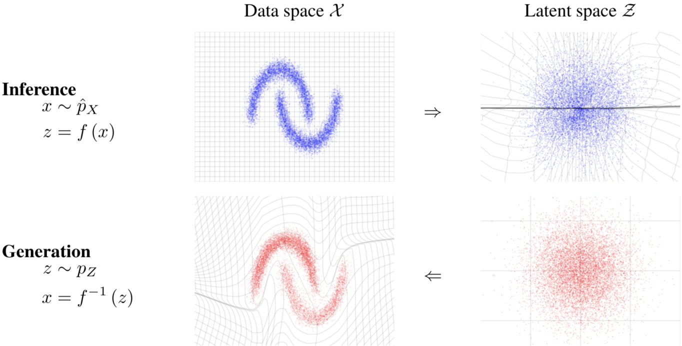
- We need to be able to compute $\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{z})$ and its Jacobian.
- We need to be able to sample from the density $p(\mathbf{z})$ (do not need to evaluate it) and to evaluate(!) $\pi(\mathbf{x})$.
- We don't need to think about computing the function $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$.

Normalizing flows KL duality

Theorem

Fitting NF model $p(\mathbf{x}|\boldsymbol{\theta})$ to the target distribution $\pi(\mathbf{x})$ using forward KL (MLE) is equivalent to fitting the induced distribution $p(\mathbf{z}|\boldsymbol{\theta})$ to the base $p(\mathbf{z})$ using reverse KL:

$$\arg\min_{\boldsymbol{\theta}} KL(\pi(\mathbf{x})||p(\mathbf{x}|\boldsymbol{\theta})) = \arg\min_{\boldsymbol{\theta}} KL(p(\mathbf{z}|\boldsymbol{\theta})||p(\mathbf{z})).$$



Papamakarios G. et al. Normalizing flows for probabilistic modeling and inference, 2019

Normalizing flows KL duality

Theorem

$$\arg\min_{\boldsymbol{\theta}} KL(\pi(\mathbf{x})||p(\mathbf{x}|\boldsymbol{\theta})) = \arg\min_{\boldsymbol{\theta}} KL(p(\mathbf{z}|\boldsymbol{\theta})||p(\mathbf{z})).$$

Proof

- $\mathbf{z} \sim p(\mathbf{z})$, $\mathbf{x} = \mathbf{g}_{\boldsymbol{\theta}}(\mathbf{z})$, $\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})$;
- $\mathbf{x} \sim \pi(\mathbf{x})$, $\mathbf{z} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$, $\mathbf{z} \sim p(\mathbf{z}|\boldsymbol{\theta})$;

$$\log p(\mathbf{z}|\boldsymbol{\theta}) = \log \pi(\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{z})) + \log|\det(\mathbf{J}_{\mathbf{g}})|;$$
$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})) + \log|\det(\mathbf{J}_{\mathbf{f}})|.$$

$$\begin{aligned} KL\left(p(\mathbf{z}|\boldsymbol{\theta})||p(\mathbf{z})\right) &= \mathbb{E}_{p(\mathbf{z}|\boldsymbol{\theta})}\left[\log p(\mathbf{z}|\boldsymbol{\theta}) - \log p(\mathbf{z})\right] = \\ &= \mathbb{E}_{p(\mathbf{z}|\boldsymbol{\theta})}\left[\log \pi(\mathbf{g}_{\boldsymbol{\theta}}(\mathbf{z})) + \log|\det(\mathbf{J}_{\mathbf{g}})| - \log p(\mathbf{z})\right] = \\ &= \mathbb{E}_{\pi(\mathbf{x})}\left[\log \pi(\mathbf{x}) - \log|\det(\mathbf{J}_{\mathbf{f}})| - \log p(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}))\right] = \\ &= \mathbb{E}_{\pi(\mathbf{x})}\left[\log \pi(\mathbf{x}) - \log p(\mathbf{x}|\boldsymbol{\theta})\right] = KL(\pi(\mathbf{x})||p(\mathbf{x}|\boldsymbol{\theta})). \end{aligned}$$

Papamakarios G., Pavlakou T., Murray I. Masked Autoregressive Flow for Density Estimation, 2017

Outline

1. Forward and Reverse KL for NF
2. Latent variable models (LVM)
3. Variational lower bound (ELBO)
4. EM-algorithm
5. Amortized inference

Bayesian framework

Bayes theorem

$$p(\mathbf{t}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{t})p(\mathbf{t})}{p(\mathbf{x})} = \frac{p(\mathbf{x}|\mathbf{t})p(\mathbf{t})}{\int p(\mathbf{x}|\mathbf{t})p(\mathbf{t})d\mathbf{t}}$$

- $\mathbf{x}$ – observed variables, $\mathbf{t}$ – unobserved variables (latent variables/parameters);
- $p(\mathbf{x}|\mathbf{t})$ – likelihood;
- $p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{t})p(\mathbf{t})d\mathbf{t}$ – evidence;
- $p(\mathbf{t})$ – prior distribution, $p(\mathbf{t}|\mathbf{x})$ – posterior distribution.

Meaning

We have unobserved variables $\mathbf{t}$ and some prior knowledge about them $p(\mathbf{t})$. Then, the data $\mathbf{x}$ has been observed. Posterior distribution $p(\mathbf{t}|\mathbf{x})$ summarizes the knowledge after the observations.

Bayesian framework

Let consider the case, where the unobserved variables $\mathbf{t}$ is our model parameters $\boldsymbol{\theta}$.

- $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ – observed samples;
- $p(\boldsymbol{\theta})$ – prior parameters distribution (we treat model parameters $\boldsymbol{\theta}$ as random variables).

Posterior distribution

$$p(\boldsymbol{\theta}|\mathbf{X}) = \frac{p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{X})} = \frac{p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}$$

If evidence $p(\mathbf{X})$ is intractable (due to multidimensional integration), we can't get posterior distribution and perform the exact inference.

Maximum a posteriori (MAP) estimation

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathbf{X}) = \arg\max_{\boldsymbol{\theta}} \bigl(\log p(\mathbf{X}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})\bigr)$$

Latent variable models (LVM)

MLE problem

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(\mathbf{X}|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{n} p(\mathbf{x}_i|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log p(\mathbf{x}_i|\boldsymbol{\theta}).$$

The distribution $p(\mathbf{x}|\boldsymbol{\theta})$ could be very complex and intractable (as well as real distribution $\pi(\mathbf{x})$).

Extended probabilistic model

Introduce latent variable $\mathbf{z}$ for each sample $\mathbf{x}$

$$p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z}); \quad \log p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) + \log p(\mathbf{z}).$$

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

Motivation

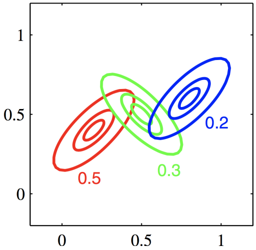
The distributions $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$ and $p(\mathbf{z})$ could be quite simple.

Latent variable models (LVM)

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

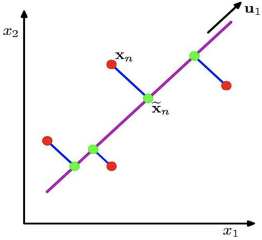
Examples

Mixture of gaussians



- $p(\mathbf{x}|z, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$
- $p(z) = \text{Categorical}(\boldsymbol{\pi})$

PCA model



- $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2 \mathbf{I})$
- $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$

Bishop C. Pattern Recognition and Machine Learning, 2006

Maximum likelihood estimation for LVM

MLE for extended problem

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{n} p(\mathbf{x}_i, \mathbf{z}_i|\boldsymbol{\theta}) =$$

$$= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log p(\mathbf{x}_i, \mathbf{z}_i|\boldsymbol{\theta}).$$

However, $\mathbf{Z}$ is unknown.

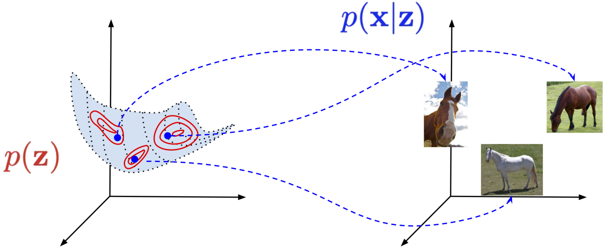
MLE for original problem

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \log p(\mathbf{X}|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log p(\mathbf{x}_i|\boldsymbol{\theta}) =$$

$$= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log \int p(\mathbf{x}_i, \mathbf{z}_i|\boldsymbol{\theta}) d\mathbf{z}_i =$$

$$= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log \int p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\theta}) p(\mathbf{z}_i) d\mathbf{z}_i.$$

Naive approach



Monte-Carlo estimation

$$p(\mathbf{x}|\boldsymbol{\theta}) = \int p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) p(\mathbf{z}) d\mathbf{z} = \mathbb{E}_{p(\mathbf{z})} p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) \approx \frac{1}{K} \sum_{k=1}^{K} p(\mathbf{x}|\mathbf{z}_k, \boldsymbol{\theta}),$$

where $\mathbf{z}_k \sim p(\mathbf{z})$.

Challenge: to cover the space properly, the number of samples grows exponentially with respect to dimensionality of $\mathbf{z}$.

image credit: https://jmtomczak.github.io/blog/4/4_VAE.html

Outline

Variational lower bound (ELBO)

Derivation 1 (inequality)

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

$$= \log \mathbb{E}_q \left[\frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} \right] \geq \mathbb{E}_q \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} = \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x})$$

Derivation 2 (equality)

$$\mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) = \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} = \int q(\mathbf{z}) \log \frac{p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x}|\boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} =$$

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

$$= \log p(\mathbf{x}|\boldsymbol{\theta}) - KL(q(\mathbf{z}) || p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}))$$

Variational decomposition

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) + KL(q(\mathbf{z}) || p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) \geq \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}).$$

Variational lower bound (ELBO)

$$\begin{aligned}\mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) &= \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} = \\ &= \int q(\mathbf{z}) \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) d\mathbf{z} + \int q(\mathbf{z}) \log \frac{p(\mathbf{z})}{q(\mathbf{z})} d\mathbf{z} \\ &= \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z})||p(\mathbf{z}))\end{aligned}$$

Log-likelihood decomposition

$$\begin{aligned}\log p(\mathbf{x}|\boldsymbol{\theta}) &= \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) + KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) \\ &= \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z})||p(\mathbf{z})) + KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})).\end{aligned}$$

- Instead of maximizing incomplete likelihood, maximize ELBO

$$\max_{\boldsymbol{\theta}} p(\mathbf{x}|\boldsymbol{\theta}) \quad \rightarrow \quad \max_{q,\boldsymbol{\theta}} \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x})$$

- Maximization of ELBO by **variational** distribution q is equivalent to minimization of KL

$$\arg\max_q \mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) \equiv \arg\min_q KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})).$$

Outline

EM-algorithm

$$\mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) = \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z},\boldsymbol{\theta}) - KL(q(\mathbf{z})||p(\mathbf{z})) =$$
$$= \mathbb{E}_q \left[\log p(\mathbf{x}|\mathbf{z},\boldsymbol{\theta}) - \log \frac{q(\mathbf{z})}{p(\mathbf{z})} \right] d\mathbf{z} \to \max_{q,\boldsymbol{\theta}}.$$

Block-coordinate optimization

- Initialize $\boldsymbol{\theta}^*$;
- **E-step** ($\mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) \to \max_q$)

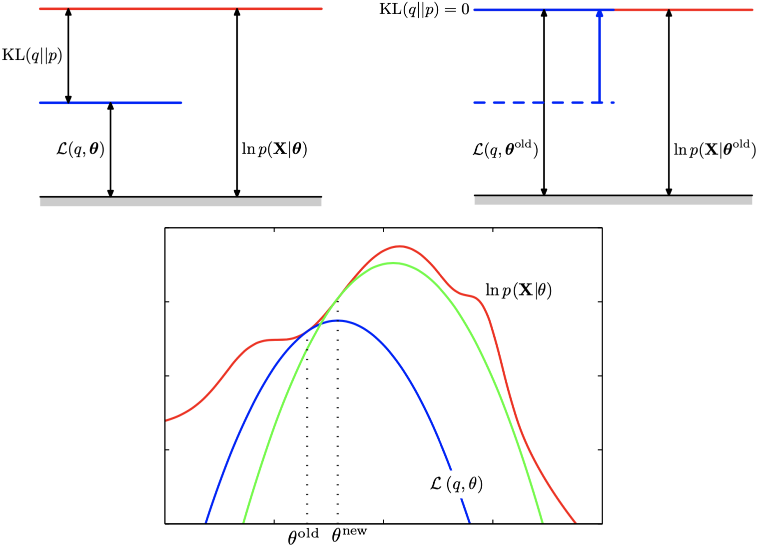
$$q^*(\mathbf{z}) = \arg\max_q \mathcal{L}_{q,\boldsymbol{\theta}^*}(\mathbf{x}) =$$
$$= \arg\min_q KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x},\boldsymbol{\theta}^*)) = p(\mathbf{z}|\mathbf{x},\boldsymbol{\theta}^*);$$

- **M-step** ($\mathcal{L}_{q,\boldsymbol{\theta}}(\mathbf{x}) \to \max_{\boldsymbol{\theta}}$)

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathcal{L}_{q^*,\boldsymbol{\theta}}(\mathbf{x});$$

- Repeat E-step and M-step until convergence.

EM-algorithm illustration



Bishop C. Pattern Recognition and Machine Learning, 2006

Outline

1. Forward and Reverse KL for NF
2. Latent variable models (LVM)
3. Variational lower bound (ELBO)
4. EM-algorithm
5. Amortized inference

Amortized variational inference

E-step

$$q(\mathbf{z}) = \arg\max_q \mathcal{L}_{q,\boldsymbol{\theta}^*}(\mathbf{x}) = \arg\min_q KL(q||p) = p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^*).$$

- $q(\mathbf{z})$ approximates true posterior distribution $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^*)$, that is why it is called **variational posterior**;
- $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^*)$ could be **intractable**;
- $q(\mathbf{z})$ is different for each object $\mathbf{x}$.

Idea

Restrict a family of all possible distributions $q(\mathbf{z})$ to a parametric class $q(\mathbf{z}|\mathbf{x}, \phi)$ conditioned on samples $\mathbf{x}$ with parameters ϕ.

Variational Bayes

- E-step

$$\boldsymbol{\phi}_k = \boldsymbol{\phi}_{k-1} + \eta \cdot \nabla_{\boldsymbol{\phi}} \mathcal{L}_{\boldsymbol{\phi}, \boldsymbol{\theta}_{k-1}}(\mathbf{x})\big|_{\boldsymbol{\phi}=\boldsymbol{\phi}_{k-1}}$$

- M-step

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \eta \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\phi}_k, \boldsymbol{\theta}}(\mathbf{x})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{k-1}}$$

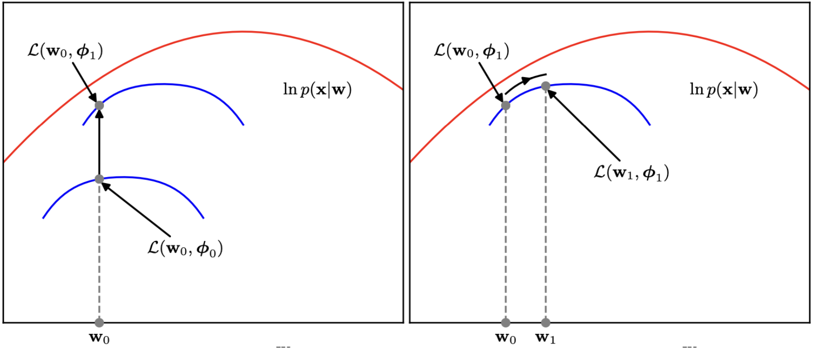
Variational EM illustration

- E-step

$$\phi_k = \phi_{k-1} + \eta \nabla_{\phi} \mathcal{L}_{\phi, \boldsymbol{\theta}_{k-1}}(\mathbf{x})\big|_{\phi=\phi_{k-1}}$$

- M-step

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\phi_k, \boldsymbol{\theta}}(\mathbf{x})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{k-1}}$$



Bishop C. Deep Learning: Foundations and Concepts, 2024

Variational EM-algorithm

ELBO

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x}) + KL(q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) \geq \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x}).$$

- **E-step**

$$\boldsymbol{\phi}_k = \boldsymbol{\phi}_{k-1} + \eta \cdot \nabla_{\boldsymbol{\phi}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}_{k-1}}(\mathbf{x})\big|_{\boldsymbol{\phi}=\boldsymbol{\phi}_{k-1}},$$

  where $\boldsymbol{\phi}$ – parameters of variational posterior distribution $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})$.

- **M-step**

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \eta \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\phi}_k,\boldsymbol{\theta}}(\mathbf{x})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{k-1}},$$

  where $\boldsymbol{\theta}$ – parameters of the generative distribution $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$.

Now all that is left is to obtain gradients: $\nabla_{\boldsymbol{\phi}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x})$, $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x})$.
Challenge: Number of samples n could be huge (we need derive the **unbiased** stochastic gradients).

Summary

- Flow duality connects data space and latent space via forward and reverse KL formulations.
- Bayesian framework is a generalization of most common machine learning tasks.
- LVM introduces latent representation of observed samples to make model more interpretative.
- LVM maximizes variational evidence lower bound (ELBO) to find MLE for the parameters.
- The general variational EM algorithm maximizes ELBO objective for LVM model to find MLE for parameters $\boldsymbol{\theta}$.
- Amortized variational inference allows to efficiently compute the stochastic gradients for ELBO using Monte-Carlo estimation.