Deep Generative Models
Lecture 2

Roman Isachenko

Ozon Masters

Spring, 2021

Recap of previous lecture

We are given i.i.d. samples $\{\mathbf{x}_i\}_{i=1}^n \in X$ (e.g. $X = \mathbb{R}^m$) from unknown distribution $\pi(\mathbf{x})$.

Goal

We would like to learn a distribution $\pi(\mathbf{x})$ for

- evaluating $\pi(\mathbf{x})$ for new samples (how likely to get object $\mathbf{x}$?);
- sampling from $\pi(\mathbf{x})$ (to get new objects $\mathbf{x} \sim \pi(\mathbf{x})$).

Challenge

Data is complex and high-dimensional.

MLE problem

Fix probabilistic model $p(\mathbf{x}|\boldsymbol{\theta})$ – a set of parameterized distributions, such that $p(\mathbf{x}|\boldsymbol{\theta}) \approx \pi(\mathbf{x})$.

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

Recap of previous lecture

Likelihood as product of conditionals

Let $\mathbf{x} = (x_1, \dots, x_m)$, $\mathbf{x}_{1:i} = (x_1, \dots, x_i)$. Then

$$p(\mathbf{x}|\boldsymbol{\theta}) = \prod_{i=1}^{m} p(x_i|\mathbf{x}_{1:i-1}, \boldsymbol{\theta}); \quad \log p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{i=1}^{m} \log p(x_i|\mathbf{x}_{1:i-1}, \boldsymbol{\theta}).$$

MLE problem for autoregressive model

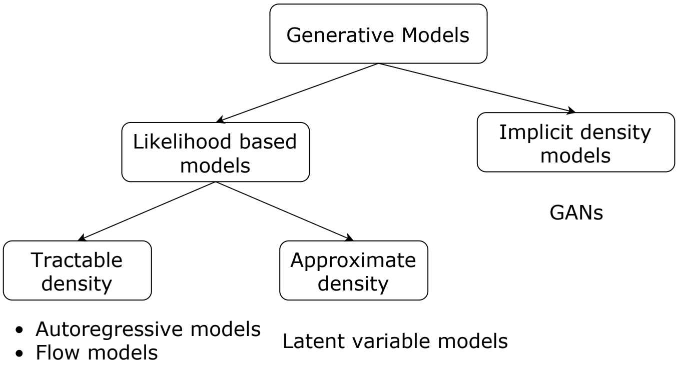
$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(\mathbf{X}|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \sum_{j=1}^{m} \log p(x_{ij}|\mathbf{x}_{i,1:j-1}\boldsymbol{\theta}).$$

Sampling

$$\hat{x}_1 \sim p(x_1|\boldsymbol{\theta}), \quad \hat{x}_2 \sim p(x_2|\hat{x}_1, \boldsymbol{\theta}), \dots, \quad \hat{x}_m \sim p(x_m|\hat{\mathbf{x}}_{1:m-1}, \boldsymbol{\theta})$$

New generated object is $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_m)$.

Generative models zoo



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 knoweldge after the obbservations.

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

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

Bayesian inference

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

Note the difference from

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

Bayesian framework

Posterior distribution

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

Bayesian inference

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

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

Maximum a posteriori (MAP) estimation

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

Bayesian framework

MAP estimation

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

Estimated $\boldsymbol{\theta}^*$ is a deterministic variable, but we could treat it as a random variable with density $p(\boldsymbol{\theta}|\mathbf{X}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$.

Dirac delta function

$$\delta(x) = \begin{cases} +\infty, & x = 0; \\ 0, & x \neq 0; \end{cases} \qquad \int f(x)\delta(x-y)dx = f(y).$$

MAP inference

$$p(\mathbf{x}|\mathbf{X}) = \int p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{X})d\boldsymbol{\theta} \approx p(\mathbf{x}|\boldsymbol{\theta}^*).$$

Latent variable models (LVM)

MLE problem

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

Challenge

$p(\mathbf{x}|\boldsymbol{\theta})$ could be intractable.

Extend 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

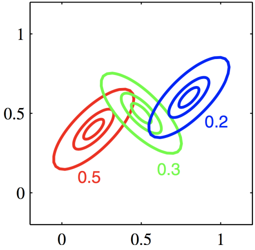
The distributions $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$ and $p(\mathbf{z})$ could be quite simple.

Latent variable models (LVM)

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

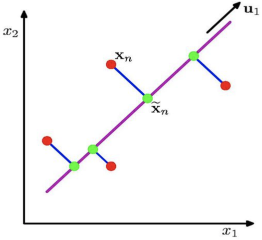
Examples

Mixture of gaussians



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

PCA model



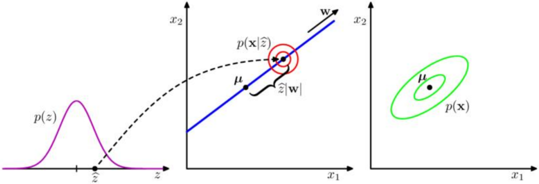
- $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \boldsymbol{\Sigma}_{\mathbf{z}})$
- $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|0, \mathbf{I})$

Bishop C. Pattern Recognition and Machine Learning, 2006

Latent variable models (LVM)

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

PCA goal: Project original data $\mathbf{X}$ onto a low dimensional latent space while maximizing the variance of the projected data.



$\boldsymbol{x} = W\boldsymbol{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}$

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

Bishop C. Pattern Recognition and Machine Learning, 2006

Incomplete likelihood

MLE

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

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

Since **Z** is unknown, maximize **incomplete likelihood**.

MILE problem

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

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

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

Variational lower bound

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

Kullback-Leibler divergence

- $KL(q||p) = \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z})} d\mathbf{z}$;
- $KL(q||p) \geq 0$;
- $KL(q||p) = 0 \Leftrightarrow q \equiv p$.

Variational lower bound

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

Evidence Lower Bound (ELBO)

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

Instead of maximizing incomplete likelihood, maximize ELBO (equivalently minimize KL)

$$\max_{\theta} p(\mathbf{x}|\boldsymbol{\theta}) \quad \rightarrow \quad \max_{q, \theta} \mathcal{L}(q, \boldsymbol{\theta}) \equiv \min_{q, \theta} KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})).$$

EM-algorithm

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

Block-coordinate optimization

- Initialize $\boldsymbol{\theta}^*$;
- E-step

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

- M-step

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

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

Ugly pic

Amortized variational inference

E-step

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

- $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^*)$ 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 particular parametric class $q(\mathbf{z}|\mathbf{x}, \phi)$ conditioned on samples $\mathbf{x}$ with parameters ϕ.

Variational Bayes

- E-step

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

- M-step

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

Variational EM-algorithm

ELBO

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

- E-step
$$\boldsymbol{\phi}_k = \boldsymbol{\phi}_{k-1} + \eta \nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}_{k-1})\big|_{\boldsymbol{\phi}=\boldsymbol{\phi}_{k-1}},$$
where ϕ – parameters of variational distribution $q(\mathbf{z}|\mathbf{x}, \phi)$.
- M-step
$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\phi}_k, \boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{k-1}},$$
where $\boldsymbol{\theta}$ – parameters of the generation function $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$.

Now all we have to do is to obtain two gradients $\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta})$, $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta})$.
Difficulty: number of samples n.

ELBO gradient (M-step, $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta})$)

$$\sum_{i=1}^{n} \mathcal{L}_i(\boldsymbol{\phi}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \mathbb{E}_q \log p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\theta}) - KL(q(\mathbf{z}_i|\mathbf{x}_i, \phi)||p(\mathbf{z}_i)) \to \max_{\boldsymbol{\phi}, \boldsymbol{\theta}}.$$

Optimization w.r.t. $\boldsymbol{\theta}$: **mini-batching** (1) + **Monte-Carlo** estimation (2)

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} \sum_{i=1}^{n} \mathcal{L}_i(\boldsymbol{\phi}, \boldsymbol{\theta}) &= \sum_{i=1}^{n} \int q(\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\phi}) \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\theta}) d\mathbf{z}_i \\
&\stackrel{(1)}{\approx} n \int q(\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\phi}) \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\theta}) d\mathbf{z}_i, \quad i \sim U[1, n] \\
&\stackrel{(2)}{\approx} n \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}_i|\mathbf{z}_i^*, \boldsymbol{\theta}), \quad \mathbf{z}_i^* \sim q(\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\phi}).
\end{aligned}$$

Monte-Carlo estimation (2):

$$\mathbb{E}_q f(\mathbf{z}) = \int q(\mathbf{z}) f(\mathbf{z}) d\mathbf{z} \approx f(\mathbf{z}^*), \text{where } \mathbf{z}^* \sim q(\mathbf{z}).$$

ELBO gradient (E-step, $\nabla_{\phi}\mathcal{L}(\phi, \boldsymbol{\theta})$)

$$\sum_{i=1}^{n} \mathcal{L}_i(\phi, \boldsymbol{\theta}) = \sum_{i=1}^{n} \mathbb{E}_q \log p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\theta}) - KL(q(\mathbf{z}_i|\mathbf{x}_i, \phi)||p(\mathbf{z}_i)) \to \max_{\boldsymbol{\phi}, \boldsymbol{\theta}}.$$

Difference from M-step: density function $q(\mathbf{z}|\mathbf{x}, \phi)$ depends on the parameters ϕ, it is impossible to use the Monte-Carlo estimation:

$$\nabla_{\phi}\mathcal{L}(\phi, \boldsymbol{\theta}) = \int \nabla_{\phi} q(\mathbf{z}|\mathbf{x}, \phi) \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) d\mathbf{z} - \nabla_{\phi} KL$$

First term is not an expectation due to the gradient.

Possible solutions

- log-derivative trick;
- reparametrization trick.

ELBO gradient (E-step, $\nabla_\phi \mathcal{L}(\phi, \boldsymbol{\theta})$)

Log-derivative trick

$$\nabla_\xi q(\eta|\xi) = q(\eta|\xi) \left(\frac{\nabla_\xi q(\eta|\xi)}{q(\eta|\xi)} \right) = q(\eta|\xi) \nabla_\xi \log q(\eta|\xi).$$

$$\nabla_\phi q(\mathbf{z}|\mathbf{x}, \phi) = q(\mathbf{z}|\mathbf{x}, \phi) \nabla_\phi \log q(\mathbf{z}|\mathbf{x}, \phi).$$

ELBO

$$\nabla_\phi \sum_{i=1}^n \mathcal{L}_i(\phi, \boldsymbol{\theta}) = \sum_{i=1}^n \int \nabla_\phi q(\mathbf{z}_i|\mathbf{x}_i, \phi) \log p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\theta}) d\mathbf{z}_i - \nabla_\phi KL$$

$$= \sum_{i=1}^n \int q(\mathbf{z}_i|\mathbf{x}_i, \phi) \left[\nabla_\phi \log q(\mathbf{z}_i|\mathbf{x}_i, \phi) \log p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\theta}) \right] d\mathbf{z}_i - \nabla_\phi KL$$

$$\approx n \nabla_\phi \log q(\mathbf{z}_i^*|\mathbf{x}_i, \phi) \log p(\mathbf{x}_i|\mathbf{z}_i^*, \boldsymbol{\theta}) - \nabla_\phi KL, \quad \mathbf{z}_i^* \sim q(\mathbf{z}_i|\mathbf{x}_i, \phi).$$

Problem

Unstable solution with huge variance.

ELBO gradient (E-step, $\nabla_{\phi}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta})$)

Reparametrization trick

$$f(\xi) = \int q(\eta|\xi)h(\eta)d\eta$$

Let $\eta = g(\xi, \epsilon)$, where g is a deterministic function, ϵ is a random variable with a density function $r(\epsilon)$.

$$\begin{aligned}\nabla_{\xi} \int q(\eta|\xi)h(\eta)d\eta = \nabla_{\xi} \int r(\epsilon)h(g(\xi, \epsilon))d\epsilon \\ \approx \nabla_{\xi} h(g(\xi, \epsilon^*)), \quad \epsilon^* \sim r(\epsilon).\end{aligned}$$

Example

$$q(\eta|\xi) = \mathcal{N}(\eta|\mu, \sigma^2), \quad r(\epsilon) = \mathcal{N}(\epsilon|0, 1), \quad \eta = \sigma \cdot \epsilon + \mu, \quad \xi = [\mu, \sigma].$$

ELBO gradient (E-step, $\nabla_\phi \mathcal{L}(\phi, \boldsymbol{\theta})$)

$$
\begin{aligned}
\nabla_\phi \sum_{i=1}^{n} \mathcal{L}_i(\phi, \boldsymbol{\theta}) = \sum_{i=1}^{n} \nabla_\phi \int q(\mathbf{z}_i|\mathbf{x}_i, \phi) \log p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\theta}) d\mathbf{z}_i - \nabla_\phi KL \\
\approx n \nabla_\phi \int r(\epsilon) \log p(\mathbf{x}_i|g(\mathbf{x}_i, \epsilon, \phi), \boldsymbol{\theta}) d\epsilon - \nabla_\phi KL, \quad i \sim U[1, n] \\
\approx n \nabla_\phi \log p(\mathbf{x}_i|g(\mathbf{x}_i, \epsilon^*, \phi), \boldsymbol{\theta}) - \nabla_\phi KL, \quad \epsilon^* \sim r(\epsilon).
\end{aligned}
$$

Variational assumption

$$
q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}), \boldsymbol{\sigma}(\mathbf{x})).
$$

$$
\mathbf{z} = g(\mathbf{x}, \epsilon, \phi) = \boldsymbol{\sigma}(\mathbf{x}) \cdot \epsilon + \boldsymbol{\mu}(\mathbf{x}).
$$

$\nabla_\phi KL(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}))$ has an analytical solution.

Variational autoencoder (VAE)

Final algorithm

- pick $i \sim U[1, n]$;
- compute a stochastic gradient w.r.t. $\boldsymbol{\phi}$

$$\begin{aligned}\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) = n \nabla_{\boldsymbol{\phi}} \log p(\mathbf{x}_i | g(\mathbf{x}_i, \boldsymbol{\epsilon}^*, \boldsymbol{\phi}), \boldsymbol{\theta}) - \\ - \nabla_{\boldsymbol{\phi}} KL(q(\mathbf{z}_i | \mathbf{x}_i, \boldsymbol{\phi}) || p(\mathbf{z}_i)), \quad \boldsymbol{\epsilon}^* \sim r(\boldsymbol{\epsilon});\end{aligned}$$

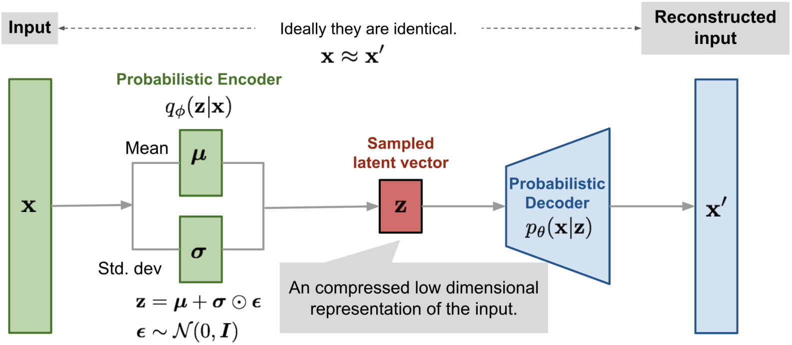
- compute a stochastic gradient w.r.t. $\boldsymbol{\theta}$

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) = n \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}_i | \mathbf{z}_i^*, \boldsymbol{\theta}), \quad \mathbf{z}_i^* \sim q(\mathbf{z}_i | \mathbf{x}_i, \boldsymbol{\phi});$$

- update $\boldsymbol{\theta}, \boldsymbol{\phi}$ according to the selected optimization method (SGD, Adam, RMSProp).

Variational autoencoder (VAE)

- Encoder $q(\mathbf{z}|\mathbf{x}, \phi) = \mathrm{NN}_e(\mathbf{x}, \phi)$ outputs $\boldsymbol{\mu}(\mathbf{x})$ and $\boldsymbol{\sigma}(\mathbf{x})$.
- Decoder $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \mathrm{NN}_d(\mathbf{z}, \boldsymbol{\theta})$ outputs parameters of the sample distribution.



---

image credit: https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

Variational Autoencoder

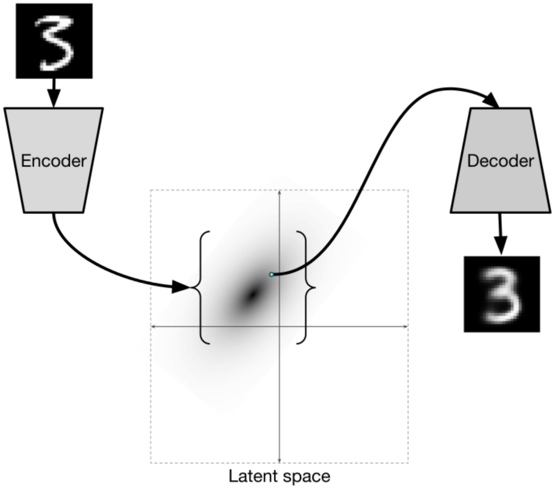


image credit: *http://ijdykeman.github.io/ml/2016/12/21/cvae.html*

Variational Autoencoder

Generation objects by sampling the latent space $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$

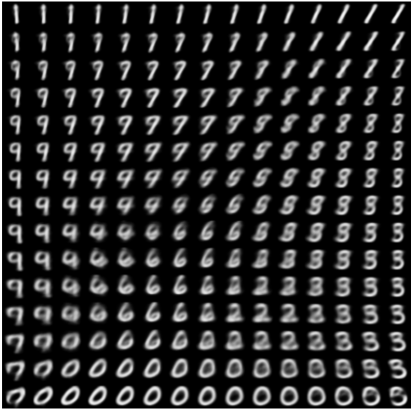


image credit: https://habr.com/ru/post/331664

Summary

- Bayesian inference is a generalization of most common machine learning tasks. It allows to construct MLE, MAP and bayesian inference, to compare models complexity and many-many more cool stuff.
- LVM introduce latent representation of observed samples to make model more interpretable.
- LVM maximizes variational evidence lower bound to find MLE of model parameters.
- ELBO maximization is performed by the general variational EM algorithm.
- Amortized inference allows to efficiently compute stochastic gradients for ELBO and to use deep neural networks for $q(\mathbf{z}|\mathbf{x}, \phi)$ and $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$.
- The VAE model is an LVM with an encoder network for $q(\mathbf{z}|\mathbf{x}, \phi)$ and a decoder network for $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$.