Latent variable models and variational inference

Dongwoo Kim

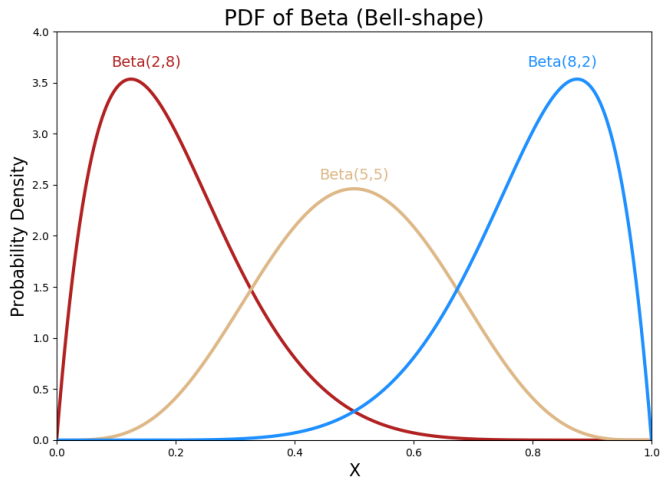
dongwoo.kim@postech.ac.kr

AI Winter School 2025

Outline

Latent Variable Models (LVMs)

- Simple distributions can be composed into a more complicated one.
- For example, the observations from a random coin flip can be considered as
  1. A coin-factory creates a coin whose mean parameter (prob. of head) is randomly sampled from Beta distribution ($0 \sim 1$).
  2. Flip coin n times and observe the number of head (Binomial distribution)



Latent Variable Models (LVMs)

1. A coin factory creates three coins; each of these coins has a different mean drawn from the Beta distribution.

$$\mu_1, \mu_2, \mu_3 \sim \text{Beta}(\alpha, \beta)$$

   where μ_1, μ_2, μ_3 are the mean parameters of the first, second, and third coins, respectively.

2. We randomly choose one coin from a uniform distribution (or some discrete distribution).

$$z_n \sim \text{Unif}(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$$

3. Flip a randomly chosen coin.

$$x_n \sim \text{Binomial}(\mu_{z_i})$$

4. Repeat 2-3 this N times.

Latent Variable Models (LVMs)

- In the previous example, variable z_i is called a latent variable.
- LV is not observable from a dataset.
- Unlike the model hyperparameters, latent variables do not parameterize the model explicitly.
- For example, which coin is used to generate x_i? A: z_i.

Formalize Latent Variable Models

- LVMs include likelihood function $p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{z})$ and the prior over $p(\boldsymbol{z})$.
- We can marginalize out $\boldsymbol{z}$ to compute the likelihood of $\boldsymbol{\theta}$ given $\boldsymbol{x}$:

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \int_{\boldsymbol{z}} p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z}$$

- MLE is defined as:

$$\max_{\theta} p(\boldsymbol{x}|\boldsymbol{\theta}) = \max_{\theta} \int_{\boldsymbol{z}} p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z}$$

- Typically, the integral is hard to compute, so more sophisticated algorithms can be used.

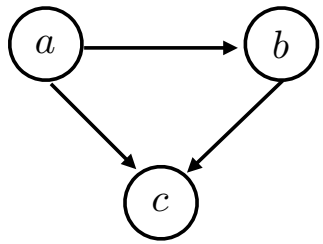
Directed Graphical Models

- In LVMs, there is explicit relationship such as $\boldsymbol{z} \rightarrow \boldsymbol{x}$.
- Can we generalize this to more complicated relationships like $\boldsymbol{z} \rightarrow \boldsymbol{x} \rightarrow \boldsymbol{y}$?
- Note that this is different from dependencies.

Directed Graphical Models

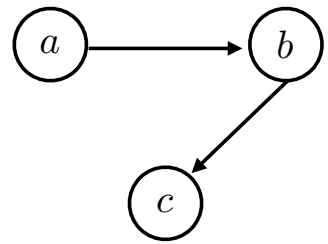
- The relation (generative assumption) between random variables can be compactly represented by a directed graph.
  - Nodes: random variables
  - Edges: relations between variables (conditional probabilities).
- Directed edge between two nodes indicates conditional probability.
- For example, if there is a link from node x to y, the distribution of y depends on x, i.e. $p(y|x)$.
- If node x doesn't have any incoming edge, then, the node is independent of the joint distribution.

Examples



The joint distribution of a, b, c is

$$p(a, b, c) = p(a)p(b|a)p(c|a, b)$$



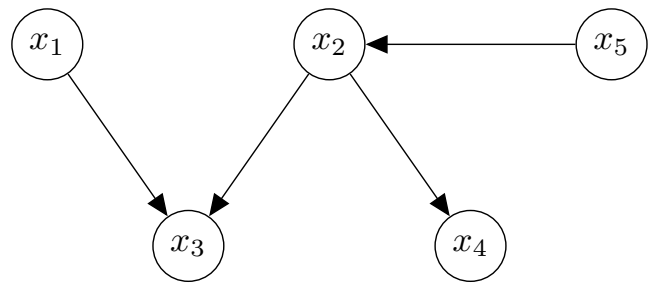
The joint distribution of a, b, c is

$$p(a, b, c) = p(a)p(b|a)p(c|b)$$

In general, the joint distribution of $p(\boldsymbol{x})$ is given as

$$p(\boldsymbol{x}) = \prod_{k=1}^{K} p(x_k | \, \mathrm{Pa}_k),$$

where Pa_k means the parent node*s* of x_k.



Given this graph, the joint distribution is defined as:

$$p(\boldsymbol{x}) = p(x_1)p(x_5)p(x_2|x_5)p(x_3|x_1, x_2)p(x_4|x_2)$$

Example: Coin Flip

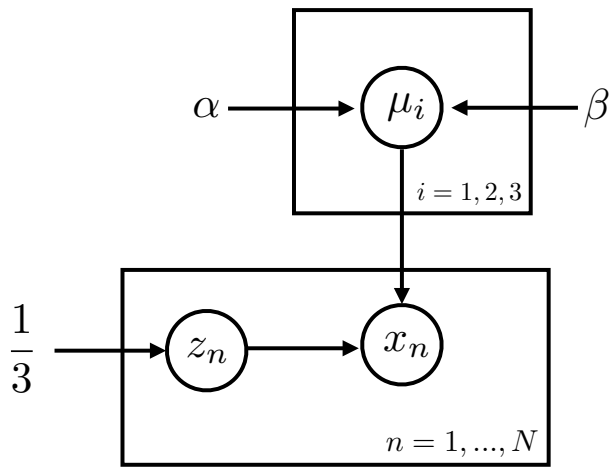


Figure: Coin flip experiments

Graphical model of the coin flip experiments

- Random variables with circle
- Hyperparameters without circle
- Plate repeats everything inside
  - 3 different μ_i
  - N different experiments z_n
  - N different experiments x_n

What would be a joint distribution of this model?

Outline

Finite Mixture Model

A semi-parametric model in the form of:

$$p(x) = \sum_{k \in [K]} p(x, z = k) = \sum_{k \in [K]} p_k(x) p(z = k)$$

- A point is drawn from one of K component distributions $\{p_k(\cdot)\}_{k \in [K]}$
- z is the latent variable indicating from which component distribution the point is originated.
- $\{\pi_k := p(z = k)\}_{k \in [K]}$ are the mixing parameters such that $\sum_{k \in [K]} \pi_k = 1$ and $\pi_k \in [0, 1]$.

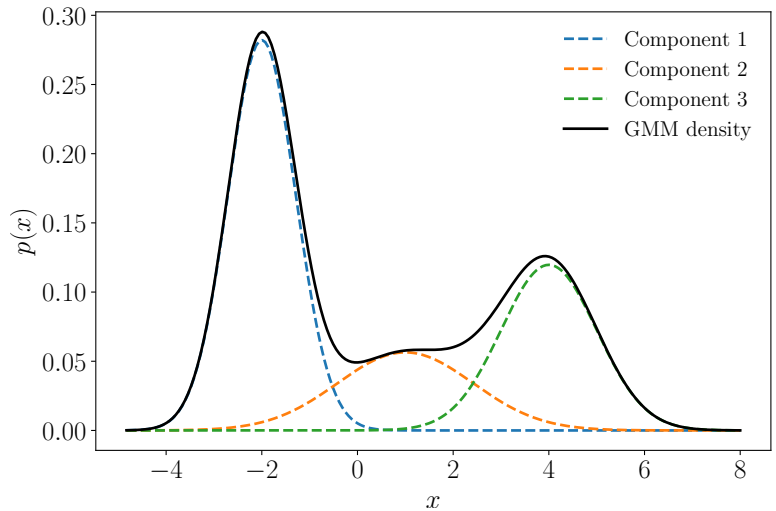
Gaussian Mixture Models: Intuition

- Marginal is a multimodal distribution

$$p_\theta(x) := p(x|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

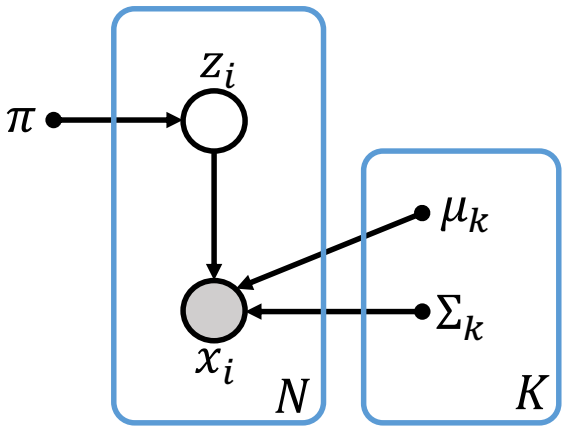
where $\theta = \{\mu_k, \Sigma_k, \pi_k\}_{k=1}^{K}$.

- Flexible than K-means.



$0.5\mathcal{N}(-2, \frac{1}{2}) + 0.2\mathcal{N}(1, 2) + 0.3\mathcal{N}(4, 1)$

Gaussian Mixture Model: Graphical Representation



Gaussian Mixture Model (GMM)

- Point x_i's true cluster $z_i \in [K]$ is hidden and independently drawn from

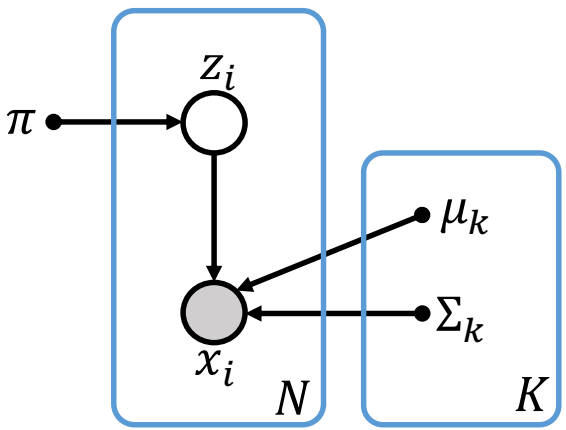
$$p(z_i) = \prod_{k \in [K]} \pi_k^{\mathbb{1}[z_i = k]} \text{, i.e., } p(z_i = k) = \pi_k.$$

- The distribution over observed variables conditioned on the latent variables is

$$p(x_i \mid z_i) = \prod_{k \in [K]} \left(\mathcal{N}(x_i \mid \mu_k, \Sigma_k)\right)^{\mathbb{1}[z_i = k]}$$

or $p(x_i \mid k) = \mathcal{N}(x_i \mid \mu_k, \Sigma_k)$ if $z_i = k$.

Learning GMM



- Compute maximum likelihood estimates of parameters

$$\theta = \{\pi_k, (\mu_k, \Sigma_k)\}_{k \in [K]}$$

- Compute the posterior on latent z_i

$$r_{ik} = p(z_i = k \mid x_i)$$

- Can optimize θ with gradient methods from marginal distribution $p(x|\theta)$?

Parameter estimation via MLE

- Given iid samples $\mathcal{D} = \{x_1, ..., x_N\}$, the log likelihood that we need to maximize is

$$
\begin{aligned}
L(\theta) = \log p(\mathcal{D}|\theta) &= \log \prod_{n=1}^{N} p(x_n|\theta) \\
&= \sum_{n=1}^{N} \log p(x_n|\theta) \\
&= \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)
\end{aligned}
$$

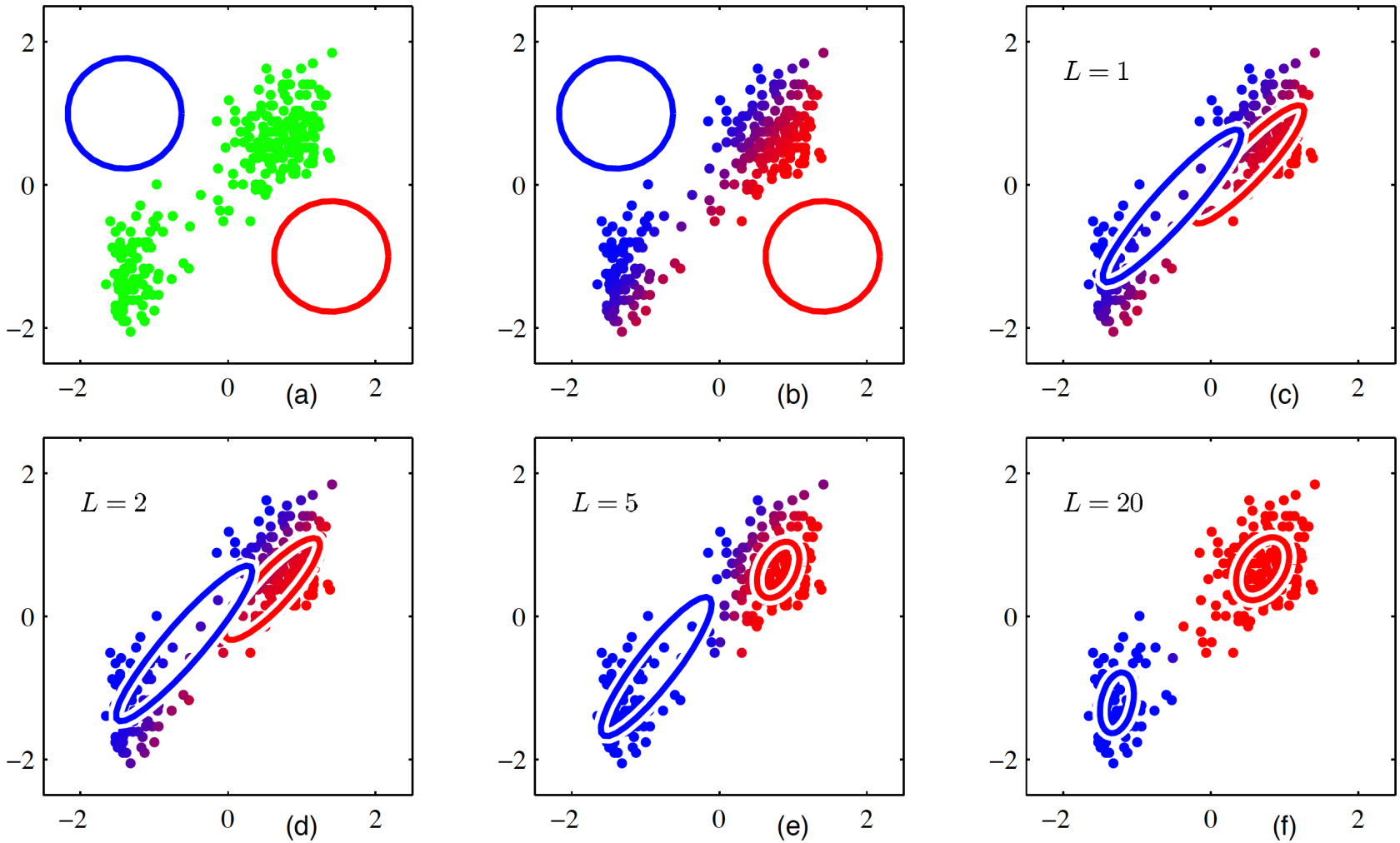
There is no Closed-form Solution

$$\sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)$$

- Can we compute $\frac{\partial L}{\partial \theta}$?
- Unfortunately, there is no closed-form solution that can find all parameters θ by a single computation.
- For example, setting the partial derivative w.r.t. π_k to zero yields

$$\frac{\partial L}{\partial \mu_k} = \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n \mid \mu_j, \Sigma_j)} \Sigma_k^{-1}(x_n - \mu_k),$$

  which cannot be simplify in a closed-form, i.e., $\pi_k =$ something.
- We will apply expectation-maximization (EM) algorithm.


2
0
−2
−2
0
2
(a)
(b)
$L = 1$
(c)
$L = 2$
(d)
$L = 5$
(e)
$L = 20$
(f)

EM for isotropic GMM

Define

- $z_i \in [K]$: the cluster to which point x_i belongs.
- θ is a set of hyperparameters, i.e., $\theta = \{\{\pi_k, \mu_k, \sigma_k\}_{k=1}^K\}$
- $\mathcal{L}_c(\theta; \{x_i, z_i\}_{i\in[N]})$: the complete-data log-likelihood, i.e.,

$$\mathcal{L}_c(\theta; \{x_i, z_i\}_{i\in[N]}) = \sum_{i\in[N]} \log p(x_i, z_i \mid \theta)$$

Isotropic GMM

- Letting $z_{ik} = \mathbb{1}[z_i = k]$,

$$p(z_i) = \prod_{k\in[K]} \pi_k^{z_{ik}}, \quad \text{and} \quad p(x_i \mid z_i) = \prod_{k\in[K]} \left(\mathcal{N}(x_i \mid \mu_k, \sigma_k^2 I)\right)^{z_{ik}}.$$

Log-Likelihood for GMM (1)

The complete-data log-likelihood can be calculated as follows:

$$
\begin{aligned}
\mathcal{L}_c(\theta; \{x_i, z_i\}_{i\in[N]}) &= \sum_{i\in[N]} \log p(x_i, z_i \mid \theta) \\
&= \sum_{i\in[N]} \log \left(p(x_i \mid z_i, \theta) p(z_i \mid \theta)\right) \\
&= \sum_{i\in[N]} \log \left(\prod_{k\in[K]} \left(\pi_k \mathcal{N}(x_i \mid \mu_k, \sigma_k^2 I)\right)^{z_{ik}} \right) \\
&= \sum_{i\in[N]} \sum_{k\in[K]} z_{ik} \log \left(\pi_k \mathcal{N}(x_i \mid \mu_k, \sigma_k^2 I)\right) .
\end{aligned}
$$

Log-Likelihood for GMM (2)

For given θ', define the responsibility[1] r_{ik} of cluster k to data point x_i,

$$r_{ik} := p(z_i = k \mid x_i; \theta') = \mathbb{E}_{z_i|x_i,\theta'}[z_{ik}] \,.$$

Then, given θ' or $\{r_{ik}\}$, the marginal log-likelihood of $\theta = \{\mu_k, \sigma_k^2\}$ can be approximated by:

$$\begin{aligned}
\mathcal{Q}(\theta; \theta') &:= \mathbb{E}_{\{z_i\}_{i\in[N]}|\{x_i\}_{i\in[N]},\theta'} \left[\mathcal{L}_c(\theta; \{x_i, z_i\}_{i\in[N]})\right] \\
&= \sum_{i\in[N]} \sum_{k\in[K]} \mathbb{E}_{z_i|x_i,\theta'} \left[z_{ik} \log\left(\pi_k \mathcal{N}(x_i \mid \mu_k, \sigma_k^2 I)\right)\right] \\
&= \sum_{i\in[N]} \sum_{k\in[K]} r_{ik} \log\left(\pi_k \mathcal{N}(x_i \mid \mu_k, \sigma_k^2 I)\right) \\
&= \sum_{i\in[N]} \sum_{k\in[K]} r_{ik} \left[\log \pi_k - \frac{D}{2} \log \sigma_k^2 - \frac{1}{2\sigma_k^2} \|x_i - \mu_k\|^2\right] + \text{const} \,.
\end{aligned}$$

[1] A posterior of z given the other parameters.

EM for GMM

Starting from an arbitrary choice of $\theta = \left\{\pi_k, \mu_k, \sigma_k^2\right\}_{k\in[K]}$,

- E-step: Compute responsibilities $\{r_{ik}\}$ for given $\theta' = \theta$:

$$r_{ik} := p(z_i = k \mid x_i; \theta') = \frac{\pi_k p(x_i \mid z_i = k, \mu_k, \sigma_k^2)}{\sum_{\ell\in[K]} \pi_\ell p(x_i \mid z_i = \ell, \mu_\ell, \sigma_\ell^2)} \ .$$

- M-step: Update θ_{new} maximizing the approximated marginal log-likelihood $\mathcal{Q}(\theta; \theta')$:

$$\theta_{\text{new}} = \arg\max_{\theta} \ \mathcal{Q}(\theta; \theta') \ .$$

M-Step: Gaussian Parameters (1)

Using the theory of optimization, we find θ such that $\nabla_\theta \mathcal{Q}(\theta) = 0$:

- Mean

$$\frac{\partial \mathcal{Q}}{\partial \mu_k} = -\frac{1}{\sigma_k^2} \sum_{i \in [N]} r_{ik}(x_i - \mu_k) = 0$$

$$\implies \mu_{k,\text{new}} = \frac{\sum_{i \in [N]} r_{ik} x_i}{\sum_{i \in [N]} r_{ik}} \ .$$

- Variance

$$\frac{\partial \mathcal{Q}}{\partial \sigma_k^2} = \sum_{i \in [N]} r_{ik} \left[-\frac{D}{\sigma_k} + \frac{1}{\sigma_k^3} \|x_i - \mu_k\|^2 \right] = 0$$

$$\implies \sigma^2_{k,\text{new}} = \frac{1}{D} \frac{\sum_{i \in [N]} r_{ik} \|x_i - \mu_{k,\text{new}}\|^2}{\sum_{i \in [N]} r_{ik}}$$

M-step: Mixing Parameter (2)

$$\mathcal{Q}(\theta) = \sum_{i\in[N]} \sum_{k\in[K]} r_{ik} \left[\log \pi_k - \frac{D}{2} \log \sigma_k^2 - \frac{1}{2\sigma_k^2} \|x_i - \mu_k\|^2 \right] + \text{const}$$

Note that $\{\pi_k\}$ must verify $\sum_{k\in[K]} \pi_k = 1$. Hence, recalling the theory of constrained optimization, consider the Lagrangian

$$\mathcal{Q}'(\theta, \lambda) = \mathcal{Q}(\theta) + \lambda \left(1 - \sum_{k\in[K]} \pi_k \right) .$$

Solving

$$\frac{\partial \mathcal{Q}'(\theta, \lambda)}{\partial \pi_k} = \sum_{i\in[N]} \frac{r_{ik}}{\pi_k} - \lambda = 0 ,$$

one can conclude that the optimal Lagrangian multiplier λ is given by $\lambda = N$, and thus

$$\pi_{k,\text{new}} = \frac{1}{N} \sum_{i\in[N]} r_{ik}$$

EM Algorithm for Isotropic GMM: Summary

Starting from an arbitrary choice of $\theta = \left\{\pi_k, \mu_k, \sigma_k^2\right\}_{k\in[K]}$,

- E-step: Compute responsibilities $\{r_{ik}\}$ for given $\theta' = \theta$:

$$r_{ik} := p(z_i = k \mid x_i; \theta') = \frac{\pi_k p(x_i \mid z_i = k, \mu_k, \sigma_k^2)}{\sum_{\ell\in[K]} \pi_\ell p(x_i \mid z_i = \ell, \mu_\ell, \sigma_\ell^2)} \,.$$

- M-step: Update θ_{new} maximizing the approximated log-likelihood:

$$\mu_{k,\text{new}} = \frac{\sum_{i\in[N]} r_{ik} x_i}{\sum_{i\in[N]} r_{ik}}$$

$$\sigma_{k,\text{new}}^2 = \frac{1}{D} \frac{\sum_{i\in[N]} r_{ik} \|x_i - \mu_{k,\text{new}}\|^2}{\sum_{i\in[N]} r_{ik}}$$

$$\pi_{k,\text{new}} = \frac{1}{N} \sum_{i\in[N]} r_{ik}$$

K-means: Special Case of EM for GMM?

Selecting $\pi_k = \frac{1}{K}$ and $\sigma_k^2 = \sigma^2$ for each $k \in [K]$ and infinitesimal $\sigma^2 \to 0$,

- E-step: Compute responsibilities $\{r_{ik}\}$ for given $\theta' = \theta$:

$$r_{ik} \to \mathbb{1}\left[k = \arg\max_{\ell \in [K]} p(x_i \mid z_i = \ell, \mu_\ell, \sigma^2)\right] .$$

- M-step: Update θ_{new} maximizing the approximated log-likelihood:

$$\mu_{k,\text{new}} = \frac{\sum_{i \in [N]} r_{ik} x_i}{\sum_{i \in [N]} r_{ik}}$$

$$\pi_{k,\text{new}} = \frac{1}{N} \sum_{i \in [N]} r_{ik}$$

Estimation with Latent Variables

When there are missing data or latent variables, denoted by z, MLE seeks to find θ maximizing the marginal likelihood of the observed data x:

$$p(x \mid \theta) = \int p(x, z \mid \theta) dz \ .$$

As such, MLE or MAP often require the computationally intractable marginalization or maximization. Variational inference is a family of techniques to approximate the marginalization or maximization, e.g.,

- Belief propagation
- Expectation-maximization
- Mean field approximation
- ...

Outline

Convex Set and Function

- A set $C \subset \mathbb{R}^d$ is convex if

$$\lambda x + (1-\lambda)y \in C\ , \quad \forall x, y \in C \text{ and } \forall \lambda \in [0,1].$$

- For a convex set $C \subset \mathbb{R}^d$, a function $f : C \mapsto \mathbb{R}$ is convex if

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)\ , \quad \forall x, y \in C \text{ and } \forall \lambda \in [0,1].$$

Jensen's Inequality

Theorem (Jensen's inequality for random variables)

For a convex set C, if function $f : C \mapsto \mathbb{R}$ is convex and X is a random vector on C, then

$$\mathbb{E}[f(X)] \geq f(\mathbb{E}[X]) \, .$$

In case of concave f, we have $\mathbb{E}[f(X)] \leq f(\mathbb{E}[X])$.

Proof of Jensen's Inequality

For simplicity, consider discrete random vector X with $p_i = p(X = x_i)$ for $\{x_i\}_{i\in[N]} \subset C$. We prove $\sum_{i\in[N]} p_i f(x_i) \geq f(\sum_{i\in[N]} p_i x_i)$ by recursion:

$$f\left(\sum_{i\in[N]} p_i x_i\right) = f\left(p_1 x_1 + (1-p_1)\left(\frac{\sum_{i=2}^{N} p_i x_i}{1-p_1}\right)\right)$$

$$\leq p_1 f(x_1) + (1-p_1) f\left(\frac{\sum_{i=2}^{N} p_i x_i}{1-p_1}\right)$$

$$= p_1 f(x_1) + (1-p_1) f\left(\frac{p_2}{1-p_1} x_2 + \left(\frac{1-\sum_{i=1}^{2} p_i}{1-p_1}\right)\left(\frac{\sum_{i=3}^{N} p_i x_i}{1-\sum_{i=1}^{2} p_i}\right)\right)$$

$$\leq p_1 f(x_1) + (1-p_1)\left(\left(\frac{p_2}{1-p_1}\right) f(x_2) + \left(\frac{1-\sum_{i=1}^{2} p_i}{1-p_1}\right) f\left(\frac{\sum_{i=3}^{N} p_i x_i}{1-\sum_{i=1}^{2} p_i}\right)\right)$$

$$= p_1 f(x_1) + p_2 f(x_2)\left(1-\sum_{i=1}^{2} p_i\right) f\left(\frac{\sum_{i=3}^{N} p_i x_i}{1-\sum_{i=1}^{2} p_i}\right) \ldots$$

Information and Entropy

- Information $I(X)$ of random variable X is defined as

$$I(X) := -\log p(X) \,,$$

  which is itself a random variable, and quantifies the surprise or uncertainty of the realization of X.

- Entropy $H(X)$ of random variable X is defined as the expected value of information:

$$H(X) := \mathbb{E}[I(X)] = -\sum_{x \in \mathcal{X}} p(x) \log_b p(x) \,,$$

  which measures the uncertainty of X w.r.t. base $b > 0$, and $\mathcal{X}$ is the set of all possible values of X.

In fact, those concepts were developed in the information theory to study communication system. The entropy $H(X)$ can be interpreted as the minimum bits to express data X.

Entropy and Relative Entropy

- Entropy is a measure of uncertainty of a random variable, defined by:

$$H(X) := \mathbb{E}[I(X)] = -\sum_{x \in \mathcal{X}} p(x) \log p(x) \, .$$

- Kullback-Leibler divergence is a measure of relative entropy of distribution p to reference distribution q such that p is absolutely continuous w.r.t. q, i.e., $q(x) = 0$ implies $p(x) = 0$, defined by:

$$\mathrm{KL}(p \| q) := \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \, ,$$

where we use the convention of $0 \log(0/0) = 0$.

Gibb's Inequality

Theorem (Gibb's Inequality)

For any distributions p, q such that $p \ll q$, i.e., p is absolutely continuous w.r.t. q,

$$KL(p\|q) \geq 0 \ ,$$

where the equality holds iff $p = q$.

Proof) Consider discrete distributions $\{p_i\}, \{q_i\}$.

$$\begin{aligned} \mathrm{KL}(p\|q) = \sum_i p_i \log \frac{p_i}{q_i} &= -\sum_i p_i \log \frac{q_i}{p_i} \\ &\geq -\log\left(\sum_i p_i \frac{q_i}{p_i}\right) \qquad \text{(Jensen's ineq.)} \\ &= -\log\left(\sum_i q_i\right) = 0 \ . \end{aligned}$$

Gibb's Inequality: Proof of the Equality

In order to find the "distribution" p which minimizes $\mathrm{KL}(p\|q)$, we consider Lagrangian

$$\mathcal{F}(p,\lambda) = \mathrm{KL}(p\|q) + \lambda\left(1 - \sum_i p_i\right) = \sum_i p_i \log\frac{p_i}{q_i} + \lambda\left(1 - \sum_i p_i\right) \ .$$

Then, the minimal p must have λ verifying:

$$\frac{\partial\mathcal{F}}{\partial p_i} = \log p_i - \log q_i + 1 - \lambda = 0 \ ,$$

which implies $p_i = q_i \exp(\lambda - 1)$ for each i.
Using $\sum_i p_i = 1 = \sum_i q_i \exp(\lambda - 1)$, it follows that $\lambda = 1$.
Hence, the minimal p should be identical to q, and $\mathrm{KL}(p\|q) = 0$ on such choice of p.

A Lower Bound on the Log-Likelihood (1)

The log-likelihood of model parameter θ given observation x is:

$$\begin{aligned}\mathcal{L}(\theta) &= \log p(x \mid \theta) \\ &= \log \int p(x, z \mid \theta) dz \ ,\end{aligned}$$

where we marginalize out the latent variables z in the second equality.
For any distribution $q(z)$ of the latent variables z, we have

$$\begin{aligned}\mathcal{L}(\theta) &= \log \left(\int q(z) \frac{p(x, z \mid \theta)}{q(z)} dz \right) \\ &\geq \int q(z) \log \left(\frac{p(x, z \mid \theta)}{q(z)} \right) dz \qquad \text{(Jensen's ineq.)} \ .\end{aligned}$$

A Lower Bound on the Log-Likelihood (2)

Denote the lower bound by $\mathcal{F}(q, \theta)$:

$$\begin{aligned}\mathcal{F}(q, \theta) &:= \int q(z) \log \left(\frac{p(x, z \mid \theta)}{q(z)} \right) dz \\ &= \int q(z) \log p(x, z \mid \theta) dz + H(q) \quad \text{(Def. of entropy)} .\end{aligned}$$

where $H(q)$ is the entropy of q.
One can design an EM algorithm using this lower bound:

- E-step: Maximize $\mathcal{F}(q, \theta)$ over q for tighter lower bound
- M-step: Maximize $\mathcal{F}(q, \theta)$ over θ to update estimates of θ.

EM Algorithm with max-max Interpretation

(for $k = 1, 2, ...$)

- E-step: Optimize $\mathcal{F}(q, \theta)$ w.r.t. the distribution q of latent variable z given parameters $\theta^{(k)}$, i.e.,

$$q^{(k+1)} = \arg\max_{q} \quad \mathcal{F}(q, \theta^{(k)}) \ .$$

- M-step: Maximize $\mathcal{F}(q, \theta)$ w.r.t. the parameters θ given the distribution $q^{(k+1)}$ of latent variable z, i.e.,

$$\begin{aligned} \theta^{(k+1)} &= \arg\max_{\theta} \quad \mathcal{F}(q^{(k+1)}, \theta) \\ &= \arg\max_{\theta} \quad \int q^{(k+1)}(z) \log p(x, z \mid \theta) dz \ , \end{aligned}$$

where $p(x, z \mid \theta)$ is the complete-data log-likelihood.

Monotonicity of EM Algorithm

The difference between the log-likelihood and the lower bound is:

$$\begin{aligned}
\mathcal{L}(\theta) - \mathcal{F}(q, \theta) &= \log p(x \mid \theta) - \int q(z) \log \left(\frac{p(x, z \mid \theta)}{q(z)} \right) dz \\
&= \log p(x \mid \theta) - \int q(z) \log \left(\frac{p(z \mid x, \theta) p(x \mid \theta)}{q(z)} \right) dz \\
&= - \int q(z) \log \left(\frac{p(z \mid x, \theta)}{q(z)} \right) dz \\
&= \mathrm{KL}(q(\cdot) \| p(\cdot \mid x, \theta)) \,,
\end{aligned}$$

which is zero only if $q(z) = p(z \mid x, \theta)$ (Gibb's ineq.). This is what E-step finds. Hence,

$$\mathcal{L}(\theta^{(k)}) \underset{\text{E-step}}{=} \mathcal{F}(q^{(k+1)}, \theta^{(k)}) \underset{\text{M-step}}{\leq} \mathcal{F}(q^{(k+1)}, \theta^{(k+1)}) \underset{\text{Jensen}}{\leq} \mathcal{L}(\theta^{(k+1)}) \,.$$

EM Algorithm

The EM algorithm seeks to find the MLE by iteratively applying:
(for $k = 1, 2, ...$)

- E-step: Define $\mathcal{Q}(\theta; \theta^{(k)})$ as the expectation of complete-data log-likelihood w.r.t. z given x and $\theta^{(k)}$:

$$\begin{aligned}\mathcal{Q}(\theta; \theta^{(k)}) &:= \mathbb{E}_{z|x,\theta^{(k)}}[\log p(x, z \mid \theta)] \\ &= \int p(z \mid x, \theta^{(k)}) \log p(x, z \mid \theta) dz \ .\end{aligned}$$

- M-step: Find the parameters that maximize:

$$\begin{aligned}\theta^{(k+1)} &:= \arg\max_{\theta} \mathcal{Q}(\theta; \theta^{(k)}) \\ &= \arg\max_{\theta} \mathcal{F}(q, \theta) - H(q) \\ &\text{(with the choice of } q(z) = p(z|x, \theta^{(k)})) \ ,\end{aligned}$$

where the term $H(q)$ is ignored since $H(q)$ is constant w.r.t. θ.

Outline

Generative Models

Training data ~ $p_{data}(x)$

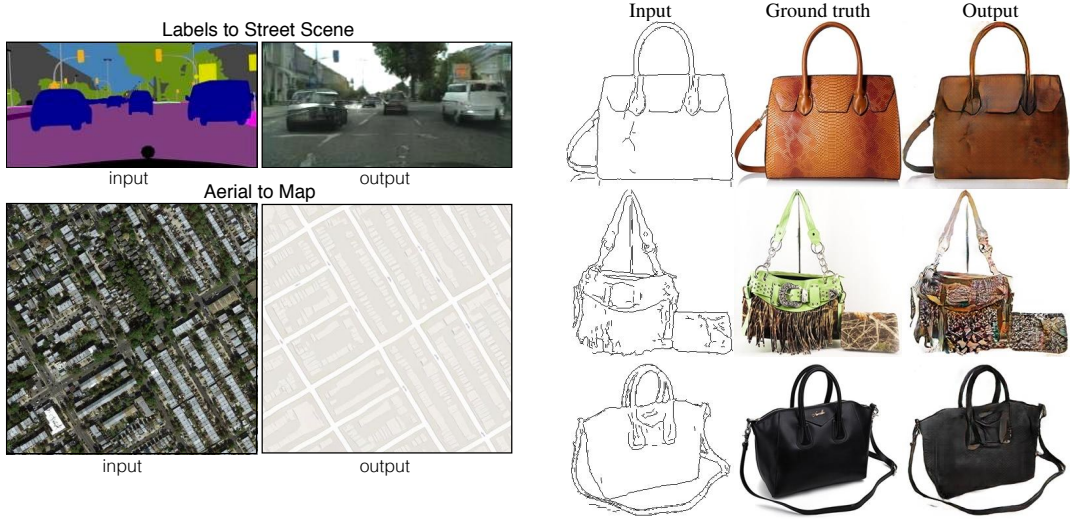
Generated samples ~ $p_{model}(x)$

Want to learn $p_{model}(x)$ similar to $p_{data}(x)$

- Given training dataset, we want to generate new samples from the same distribution
- In other words, we want to estimate density of data

Applications of Generative Model

- Generative model often provides useful approaches for other machine learning tasks, e.g., density estimation for regression, classification, out-of-distribution detection, ...
- Beside this, there are a number of interesting applications
  - e.g., Image-to-Image translation: super-resolution, style change, colorization, ...



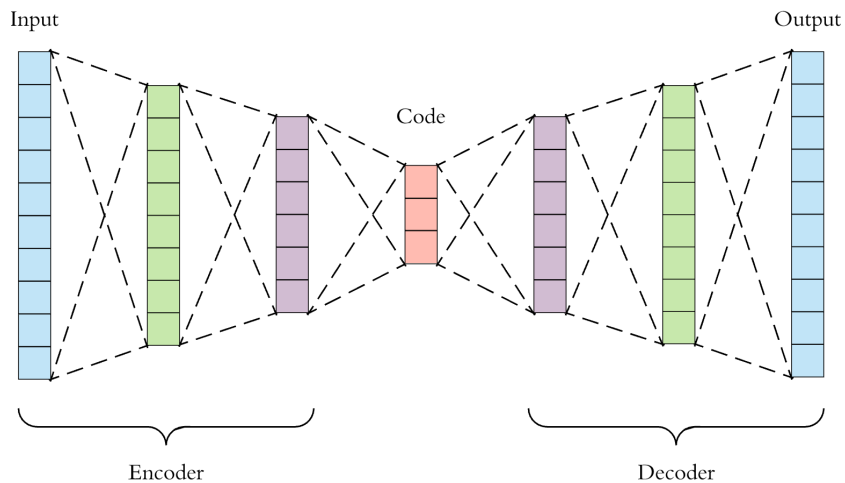
[Isola et al 16]

Applications of Generative Model

- Beside this, there are a number of interesting applications
  - e.g., POSCO steel plate with random pattern
  - More examples can be found at https://machinelearningmastery.com/impressive-applications-of-generative-adversarial-networks/

Ordinary Auto-encoder (AE) for Data Generation?

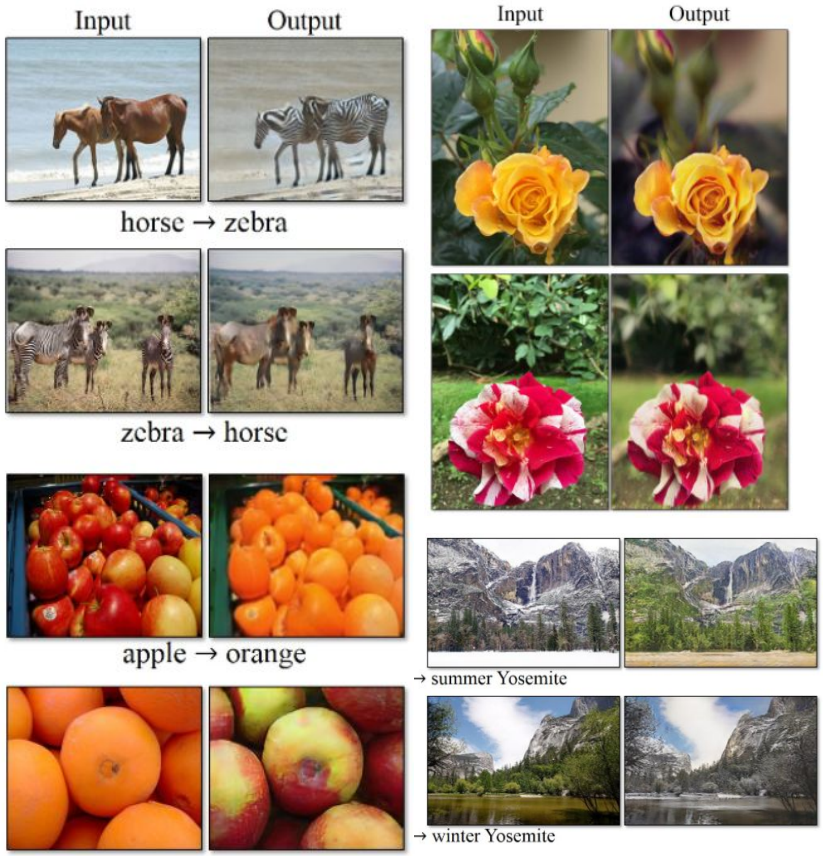
- We can use the ordinary autoencoder (in the previous lecture) to generate data
- However, we cannot control the shape of distribution of latent variable in AE
- Hence, it is hard to interpret or manipulate latent feature to generate desired data, i.e., we need some probabilistic approach



What we can do from manipulating latent feature?

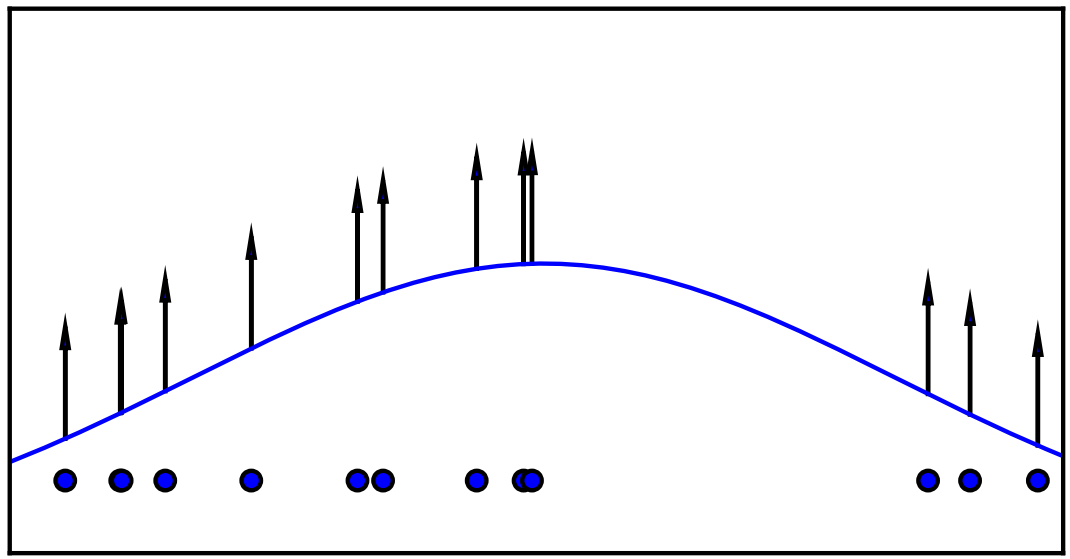
What we can do from manipulating latent feature?

Source->Target domain transfer



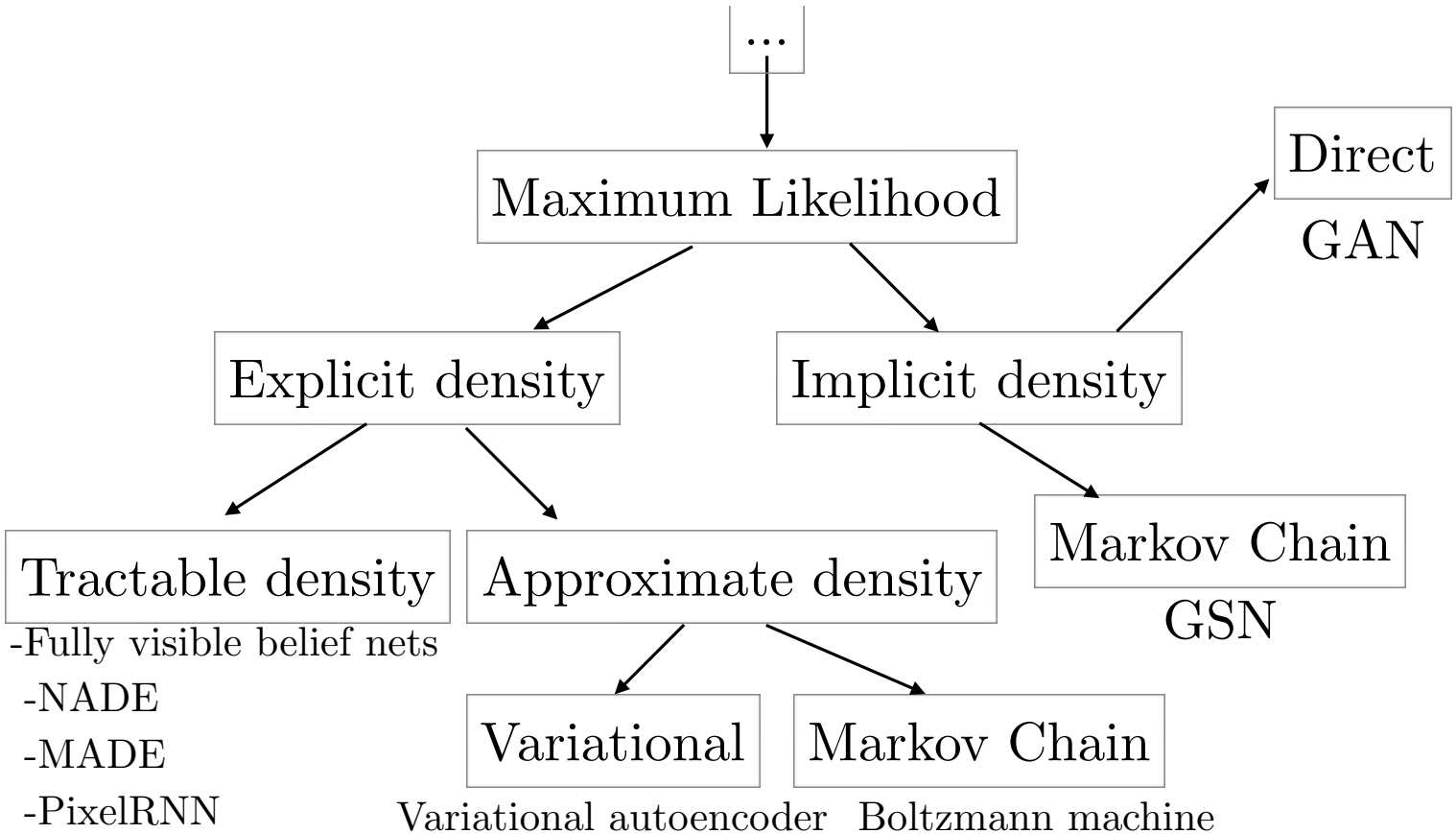
CycleGAN. Zhu et al. 2017.

A General Approach: Maximum Likelihood



$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(\boldsymbol{x} \mid \boldsymbol{\theta})$$

Taxonomy of Generative Models

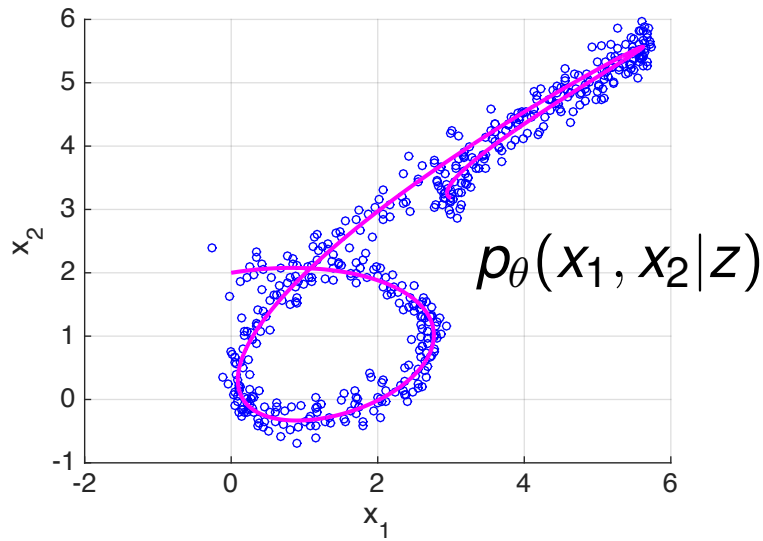


(Goodfellow 2016)

Manifold Hypothesis

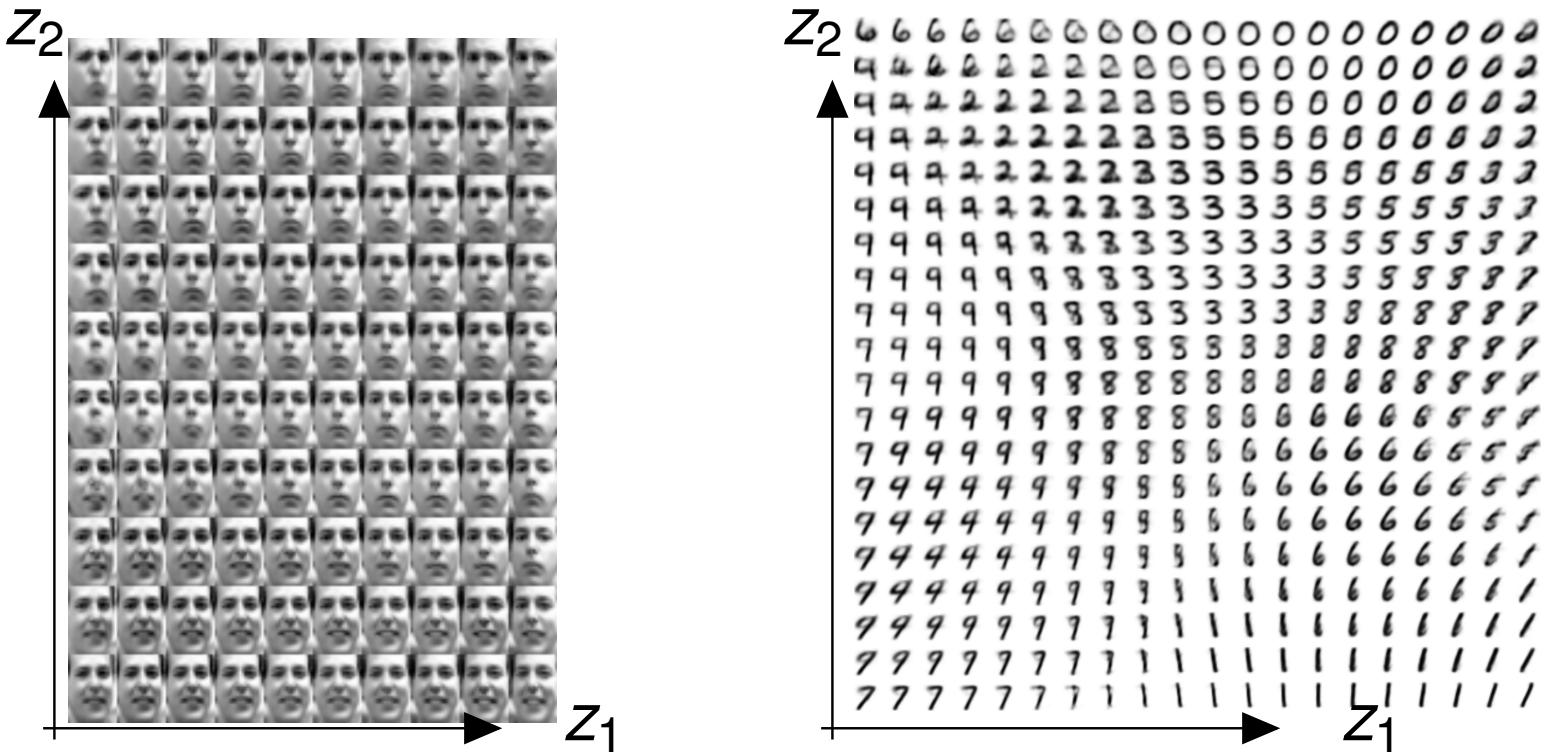
- x is a high dimensional vector
- Data is concentrated around a low dimensional manifold

$$z \in [0, 1] \quad \rightarrow$$



Manifold Hypothesis

- $x \in \mathbb{R}^D$ is a high dimensional vector
- Data is concentrated around a low dimensional manifold ($z \in \mathbb{R}^M$ with $M \ll D$)



[Kingma and Welling 14]

A Probabilistic Approach for Generative Model

Recalling MLE, our objective is maximizing

$$p_\theta(x) = \int p(z)p_\theta(x \mid z)dz$$

where generative model is $p_\theta(x \mid z)$

- Recalling manifold hypothesis, choose prior $p(z)$ to be simple, e.g., Gaussian distribution of reasonable latent attributes z, e.g., pose, degree of smile, ...
- As conditional $p_\theta(x \mid z)$ is anticipated to be complex, neural network is widely selected
- The marginalization $\int$ is intractable $\rightarrow$ variational inference

Intractability

- Data likelihood is intractable due to the integral:

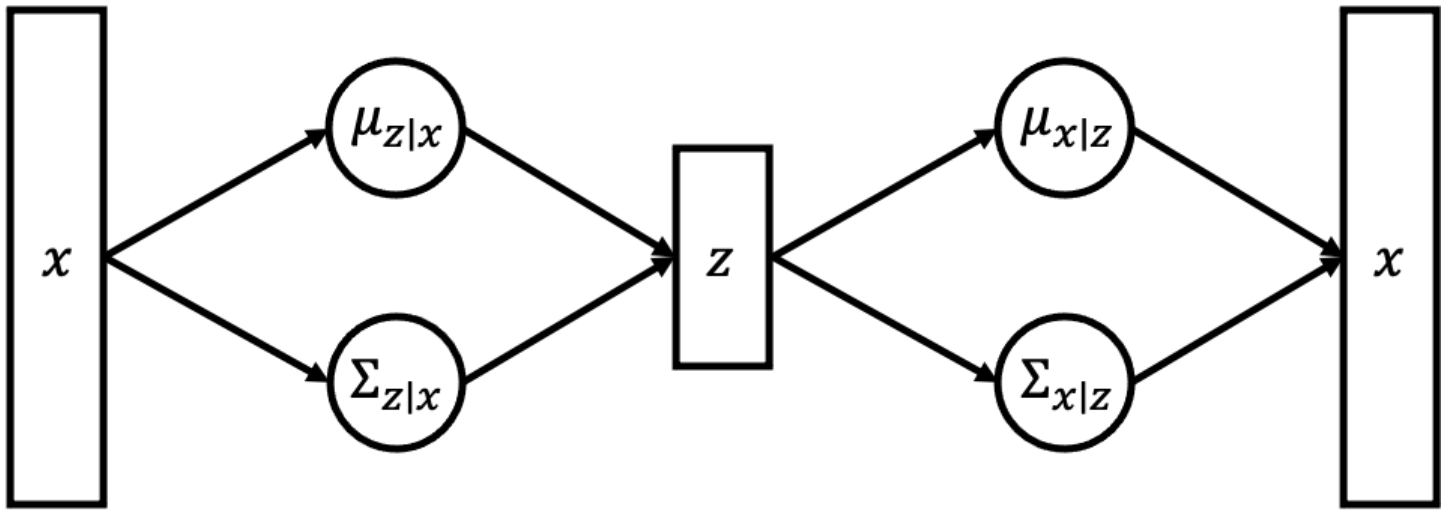
$$p_\theta(x) = \int p(z) p_\theta(x \mid z) dz$$

- Posterior density is also intractable due to the data likelihood:

$$p_\theta(z \mid x) = \frac{p_\theta(x \mid z) p(z)}{p_\theta(x)}$$

- A solution: approximate the posterior $p_\theta(z \mid x)$ using another (encoder) network $q_\phi(z \mid x)$

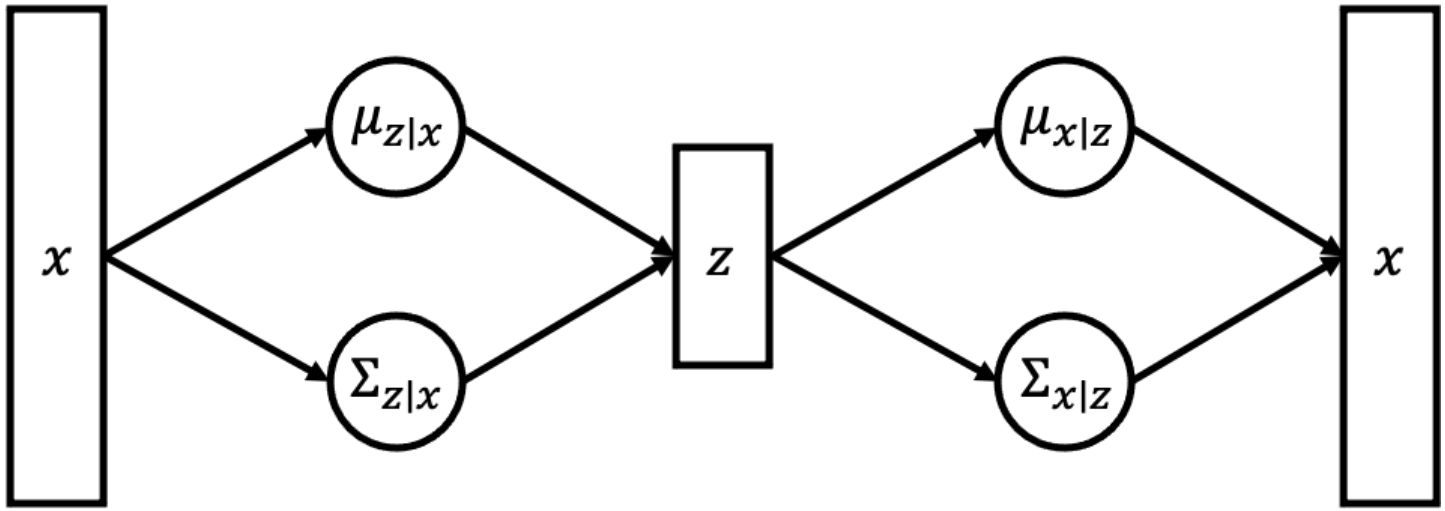
A Probabilistic Framework of Auto-encoder



You can imagine that there is a neural network f_ϕ for encoder, so that

$$f_\phi(x) = \begin{bmatrix} \mu_{z|x} \\ \sigma_{z|x} \end{bmatrix}, \quad \text{and} \quad q_\phi(z|x) = \mathcal{N}\left(f_\phi(x)_1, (f_\phi(x)_2)^2\right)$$

A Probabilistic Framework of Auto-encoder



encoder $q_\phi(z \mid x) = \mathcal{N}(\mu_{z|x}, \Sigma_{z|x})$

decoder $p_\theta(x \mid z) = \mathcal{N}(\mu_{x|z}, \Sigma_{x|z})$

For decoder p_θ,

$$g_\theta(z) = \begin{bmatrix} \mu_{x|z} \\ \sigma_{x|z} \end{bmatrix}, \quad \text{and} \quad p_\theta(x|z) = \mathcal{N}\left(g_\theta(z)_1, (g_\theta(z)_2)^2\right)$$

Variational Autoencoder

Recalling we aim at MLE: for given x [2],

$$
\begin{aligned}
\log p_\theta(x) &= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log p_\theta(x)\right] \\
&= \mathbb{E}_z\left[\log \frac{p_\theta(x \mid z)p(z)}{p_\theta(z \mid x)}\right] \\
&= \mathbb{E}_z\left[\log \frac{p_\theta(x \mid z)p(z)}{p_\theta(z \mid x)}\frac{q_\phi(z \mid x)}{q_\phi(z \mid x)}\right] \\
&= \mathbb{E}_z\left[\log p_\theta(x \mid z)\right] - \mathbb{E}_z\left[\log \frac{q_\phi(z \mid x)}{p(z)}\right] + \mathbb{E}_z\left[\log \frac{q_\phi(z \mid x)}{p_\theta(z \mid x)}\right] \\
&= \mathbb{E}_z\left[\log p_\theta(x \mid z)\right] - \mathrm{KL}(q_\phi(z \mid x)\|p(z)) + \mathrm{KL}(q_\phi(z \mid x)\|p_\theta(z \mid x))
\end{aligned}
$$

where the KL divergences take the expectation w.r.t. $z \sim q_\phi(\cdot \mid x)$.

[2] $p(x)p(z \mid x) = p(x \mid z)p(z)$

Variational Autoencoder

Recalling we aim at MLE: for given x,

$$\log p_\theta(x) = \underbrace{\mathbb{E}_z\left[\log p_\theta(x \mid z)\right]}_{\text{(A)}} - \underbrace{\mathrm{KL}(q_\phi(z \mid x) \| p(z))}_{\text{(B)}} + \underbrace{\mathrm{KL}(q_\phi(z \mid x) \| p_\theta(z \mid x))}_{\text{(C)}}$$

- Term (A) is tractable as we can sample $z \sim q_\phi(\cdot \mid x)$ from the encoder, and compute $p_\theta(x \mid z)$ from the decoder.
- Term (B) is tractable as the KL divergence between Gaussians has a closed-form
- Term (C) is intractable, while we know it is non-negative thanks to Gibbs' inequality ($KL \geq 0$)
- Hence, define (A)+(B) as variational lower bound $\mathcal{L}(x, \theta, \phi)$ (ELBO: Evidence Lower BOund [3]) and maximize it

[3] c.f. EM slides p. 12

Training VAE

Training VAE:

$$\arg\max_{\theta,\phi} \sum_{i=1}^{N} \mathcal{L}(x^{(i)}, \theta, \phi)$$

Understanding ELBO:

$$\begin{aligned}\log p_\theta(x) &\geq \mathcal{L}(x, \theta, \phi) \\ &= \mathbb{E}_z \left[\log p_\theta(x \mid z)\right] - \mathrm{KL}(q_\phi(z \mid x) \| p_\theta(z))\end{aligned}$$

- $\mathbb{E}_z \left[\log p_\theta(x \mid z)\right]$ for reconstruction
- $\mathrm{KL}(q_\phi(z \mid x) \| p(z))$ for regularization to make the approximate posterior close to the prior

Training VAE: Monte Carlo Method

Let's simplify the model by assuming $x, z \in \mathbb{R}$,

$$q_\phi(z|x) \sim \mathcal{N}(z \mid f_\phi(x), \sigma_z^2) \quad \text{and} \quad p_\theta(x|z) \sim \mathcal{N}(x|g_\theta(z), \sigma_x^2)$$

where $f_\phi(x)$ is a function of x parameterized by ϕ, and $g_\theta(z)$ is a function of z parameterized by θ.

The first term of ELBO has no analytic solution:

$$\mathbb{E}_z \left[\log p_\theta(x \mid z)\right] = \int q_\phi(z|x) \log p_\theta(x \mid z) dz$$

We can approximate the expectation with Monte-Carlo method:

$$\mathbb{E}_z \left[\log p_\theta(x \mid z)\right] \approx \frac{1}{N} \sum_{i=1}^{N} \log p_\theta(x \mid z^{(i)})$$

where $z^{(1)}, z^{(2)}, ..., z^{(N)}$ are samples drawn from $q_\phi(z|x)$.

Training Decoder p

Given the Monte Carlo approximation

$$\mathbb{E}_z \left[\log p_\theta(x \mid z)\right] \approx \frac{1}{N} \sum_{i=1}^{N} \log p_\theta(x \mid z^{(i)})$$

$$= -\log \sigma_x^2 \sqrt{2\pi} - \frac{1}{2N} \sum_{i=1}^{N} \frac{(x - g_\theta(z^{(i)}))^2}{\sigma_x^2}$$

we can approximate the derivative w.r.t θ[4].

For example, if $g_\theta(z) = \theta_1 z + \theta_0$ where $\theta_1, \theta_0 \in \mathbb{R}$,

$$\frac{\partial \, \mathbb{E}_z \left[\log p_\theta(x \mid z)\right]}{\partial \theta_1} \approx \frac{1}{N} \sum_{i=1}^{N} \frac{(x - g_\theta(z^{(i)}))z^{(i)}}{\sigma_x^2}$$

[4] The same procedure can be applied if g_θ is a NN parameterized by θ.

Training Encoder q

Again, given the Monte Carlo approximation

$$\mathbb{E}_z \left[\log p_\theta(x \mid z)\right] \approx \frac{1}{N} \sum_{i=1}^{N} \log p_\theta(x \mid z^{(i)})$$

$$= -\log \sigma_x^2 \sqrt{2\pi} - \frac{1}{2N} \sum_{i=1}^{N} \frac{(x - g_\theta(z^{(i)}))^2}{\sigma_x^2}$$

we cannot approximate the derivative w.r.t ϕ in this case.

Why? the distribution q is replaced by its samples!

$\Rightarrow$ Reparameterization is a key trick to train VAE!

Reparameterization Trick

Some random variables can be represented as a function of another variable.

For example, assume $Z \sim \mathcal{N}(\mu, \sigma^2)$.

The distribution of Z can be explained by the standard normal distribution as

$$Z = \sigma\epsilon + \mu, \quad \text{where} \quad \epsilon \sim \mathcal{N}(0, 1)$$

We can also take a sample of Z using the sample from $\mathcal{N}(0, 1)$ via $z^{(i)} = \sigma\epsilon^{(i)} + \mu$

Training Encoder with Reparameterization

Recall $q_\phi(z|x) \sim \mathcal{N}(z \mid f_\phi(x), \sigma_z^2)$.

Using reparam $z^{(i)} = \epsilon^{(i)}\sigma_z + f_\phi(x)$, the expectation can be rewritten as

$$\begin{aligned}
\mathbb{E}_z\left[\log p_\theta(x \mid z)\right] &\approx \frac{1}{N}\sum_{i=1}^{N} \log p_\theta(x \mid z^{(i)}) \\
&= -\log \sigma_x^2\sqrt{2\pi} - \frac{1}{2N}\sum_{i=1}^{N}\frac{(x - g_\theta(z^{(i)}))^2}{\sigma_x^2} \\
&= -\log \sigma_x^2\sqrt{2\pi} - \frac{1}{2N}\sum_{i=1}^{N}\frac{(x - g_\theta(\epsilon^{(i)}\sigma_z + f_\phi(x)))}{\sigma_x^2}
\end{aligned}$$

Then the partial derivative w.r.t. ϕ can be computed via

$$\frac{\partial\, \mathbb{E}_z\left[\log p_\theta(x \mid z)\right]}{\partial \phi} \approx \frac{1}{N}\sum_{i=1}^{N}\frac{(x - g_\theta(z^{(i)}))}{\sigma_x^2}\frac{\partial g_\theta}{\partial \phi}$$

KL Divergence

The second term in ELBO, i.e., $\mathrm{KL}(q_\phi(z \mid x) \| p(z))$, has an analytic solution if both q and p follows the normal distribution:

$$\int q_\theta(z|x) \log p(z) dz = \int \mathcal{N}(z; \mu, \sigma^2) \log \mathcal{N}(z; 0, I) dz$$
$$= -\frac{J}{2} \log 2\pi - \frac{1}{2} \sum_{j=1}^{J} (\mu_j^2 + \sigma_j^2)$$

where J is a dimensionality of z, μ and σ is a function of x.

We can easily compute the derivatives w.r.t μ and σ.

Training VAE: Summary

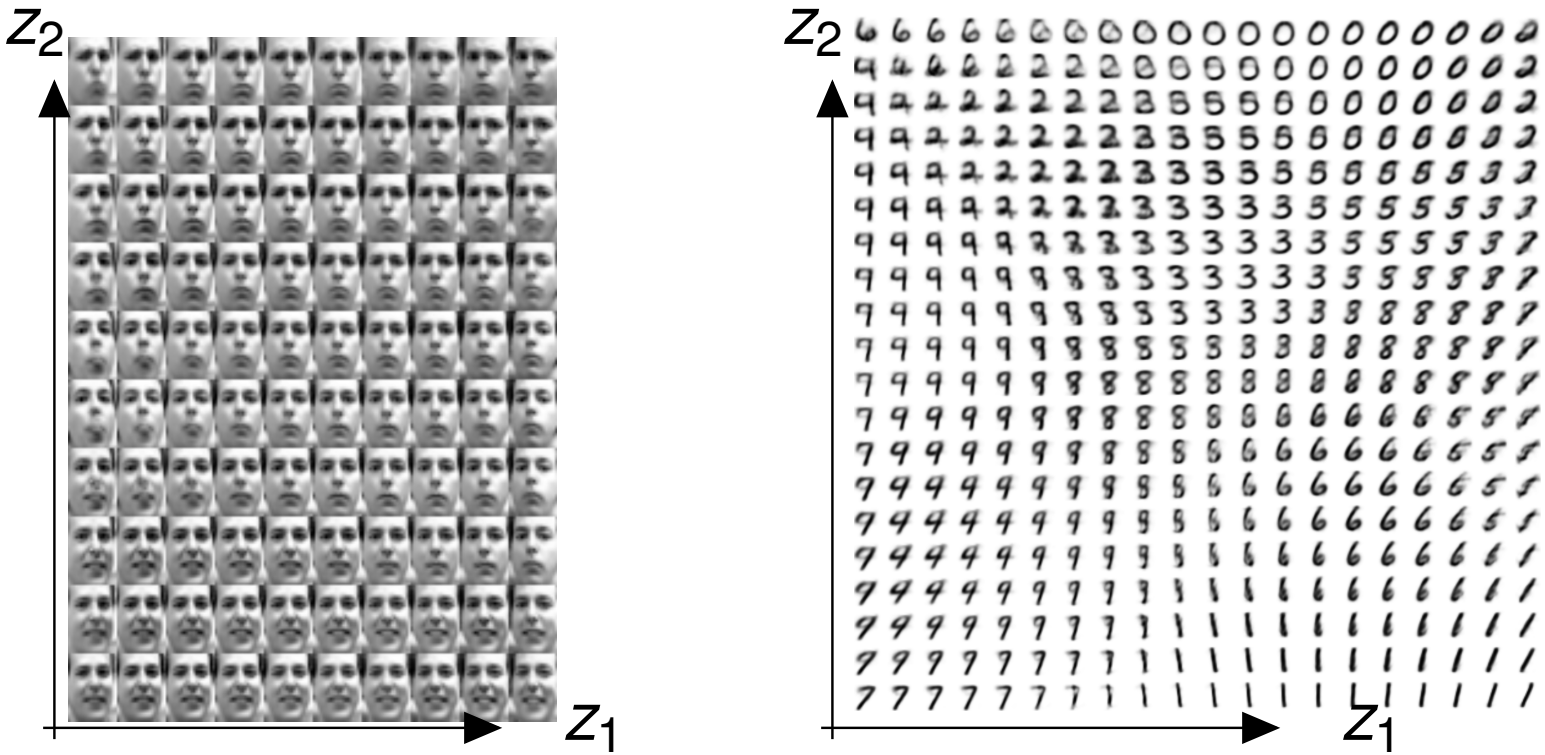
Training VAE via ELBO:

$$\begin{aligned}\log p_\theta(x) &\geq \mathcal{L}(x, \theta, \phi) \\ &= \mathbb{E}_z \left[\log p_\theta(x \mid z)\right] - \mathrm{KL}(q_\phi(z \mid x) \| p(z)) \\ &\approx \frac{1}{N} \sum_{n=1}^{N} \left[\log p_\theta(x \mid z^{(n)})\right] - \mathrm{KL}\left(q_\phi(z^{(n)} \mid x) \| p(z^{(n)})\right)\end{aligned}$$

- $\partial \mathcal{L}(x, \theta, \phi)/\partial \theta$ is simple given samples from $q(z|x)$
- $\partial \mathcal{L}(x, \theta, \phi)/\partial \phi$ requires reparameterization trick.

Generating Data from VAE

Use the decoder network with z sampled from prior $\mathcal{N}(0, I)$



[Kingma and Welling 14]

- Similar z implies similar output x
- It is interesting to see that in the left, $z_1 \approx$ head pose, and $z_2 \approx$ degree of smile