

Latent variable models and variational inference

Dongwoo Kim

dongwoo.kim@postech.ac.kr

AI Winter School 2025



⁰ Slides available at <https://tinyurl.com/22vazp6w>

Outline

- 1 Latent Variable Models
- 2 Example: Gaussian Mixture Model (GMM)
Expectation Maximization
- 3 Analysis and generalization of EM algorithm
Jensen's inequality and Gibb's inequality
Mean-Field Variational Approximation
Example with Logistic Regression
- 4 Variational Auto Encoders (VAEs)

Latent Variable Models (LVMs)

- ▶ **Definition:** LVMs introduce hidden (latent) variables to model complex data distributions.
- ▶ **Motivation:** Observed data can often be better explained by assuming underlying hidden factors.

Mathematical Formulation:

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

where:

- ▶ x is the observed variable
- ▶ z is the latent variable
- ▶ θ are model parameters

Latent Variable Models (LVMs)

Challenges:

- ▶ *Posterior Inference:*

$$p(z | x, \theta) = \frac{p(x, z | \theta)}{p(x | \theta)} = \frac{p(x | z, \theta) p(z | \theta)}{\int p(x | z', \theta) p(z' | \theta) dz'}$$

is often *intractable*.

- ▶ The integral or summation over z complicates direct analytical solutions.
- ▶ Requires *approximate methods* (e.g., Expectation-Maximization or Variational Inference) to estimate $p(z | x, \theta)$.

Example: Customer Segmentation in Retail

Scenario:

- ▶ A retail store tracks every customer's purchase history over time.
- ▶ *Observed data:* Which items each customer buys and how frequently.
- ▶ *Latent variable:* Each customer's **segment** or **cluster** (e.g., "budget-conscious," "brand-focused," "tech-savvy," etc.).

Simple Model (Mixture Idea):

$$z_i \sim \text{Discrete}(\pi_1, \pi_2, \dots, \pi_K), \\ \mathbf{x}_i \mid z_i \sim \mathcal{F}(\theta_{z_i}),$$

where:

- ▶ z_i indicates the (hidden) segment of customer i .
- ▶ \mathcal{F} is a probability model for purchase behaviors (e.g., Poisson, multinomial), parameterized by θ_{z_i} .
- ▶ π_k are segment mixing weights ($\sum_k \pi_k = 1$).

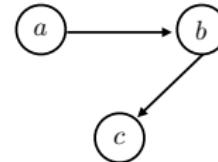
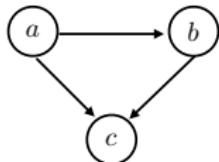
Directed Graphical Models

- ▶ In LVMs, there is explicit relationship such as $z \rightarrow x$.
- ▶ Can we generalize this to more complicated relationships like $z \rightarrow x \rightarrow y$?
- ▶ 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

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

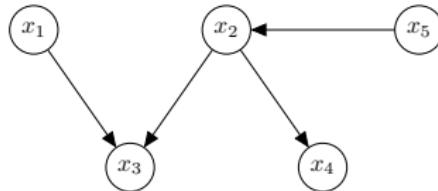
The joint distribution of a, b, c is

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

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

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

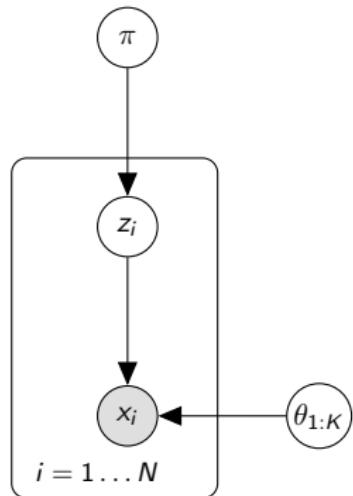
where Pa_k means the parent nodes of x_k .



Given this graph, the joint distribution is defined as:

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

Graphical Model: Customer Segmentation



- ▶ Random variables with circle
- ▶ Hyperparameters without circle
- ▶ Plate repeats everything inside
- ▶ Graphical model of the customer segmentation
 - ▶ For each customer i :
 - ▶ Draw latent $z_i \sim \pi$
 - ▶ Observe $x_i \sim \mathcal{F}(\theta_{z_i})$

Outline

- 1 Latent Variable Models**
- 2 Example: Gaussian Mixture Model (GMM)**
Expectation Maximization
- 3 Analysis and generalization of EM algorithm**
Jensen's inequality and Gibb's inequality
Mean-Field Variational Approximation
Example with Logistic Regression
- 4 Variational Auto Encoders (VAEs)**

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 $\pi_k \in [0, 1]$.

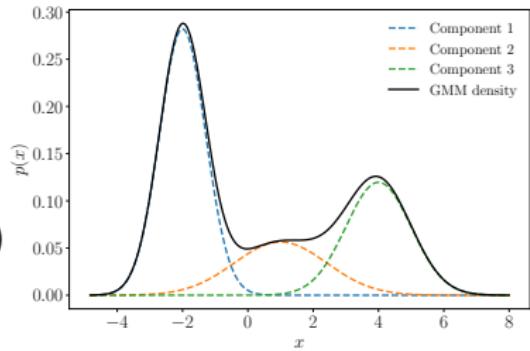
Gaussian Mixture Models: Intuition

- 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.



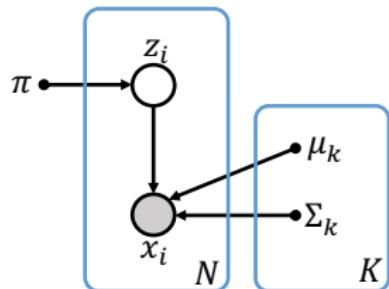
$$0.5\mathcal{N}(-2, \frac{1}{2}) + 0.2\mathcal{N}(1, 2) + 0.3\mathcal{N}(4, 1)$$

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

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

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

Learning GMM

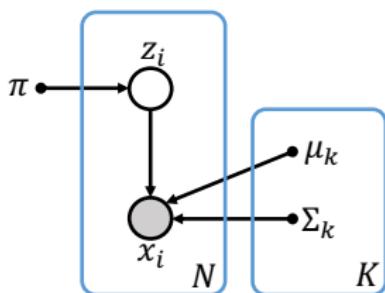
- ▶ Compute maximum likelihood estimates of parameters

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

- ▶ Compute the posterior on latent z_i

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

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



Parameter estimation via MLE

- Given iid samples $\mathcal{D} = \{x_1, \dots, x_N\}$, 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

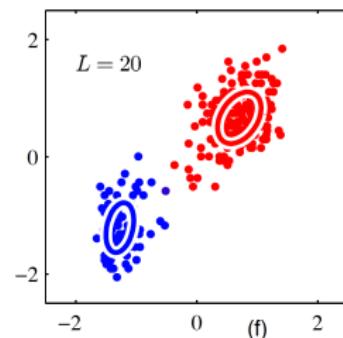
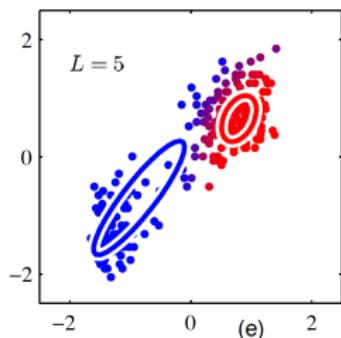
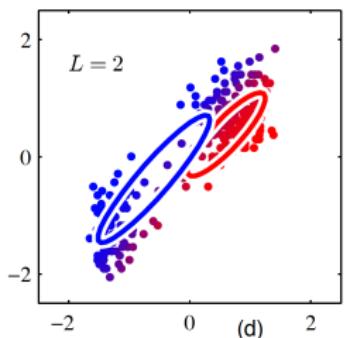
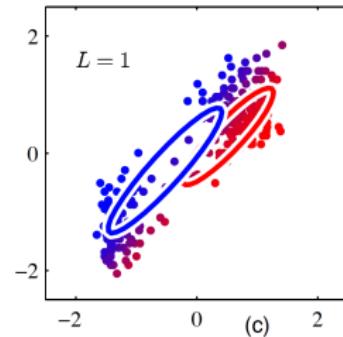
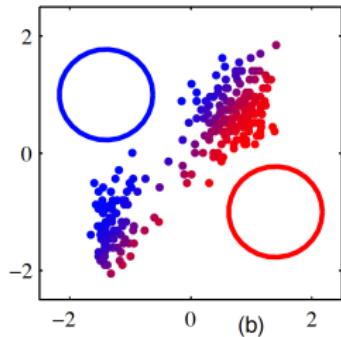
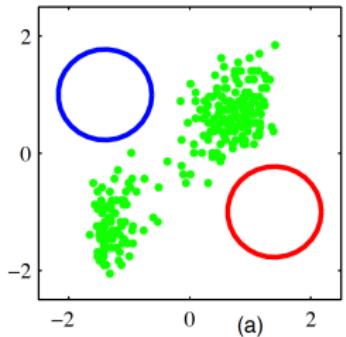
$$L(\theta) = \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)$$

- ▶ 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

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

which cannot be simplified in a closed form, i.e., $\pi_k = \text{something}$.

- ▶ We will apply the expectation-maximization (EM) algorithm.



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

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

Log-Likelihood for GMM (1)

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

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

Log-Likelihood for GMM (2)

For given θ' , define the responsibility¹ 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:

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

¹A posterior of z given the other parameters.

EM for GMM

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

- ▶ 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

$$\begin{aligned}\frac{\partial \mathcal{Q}}{\partial \mu_k} &= -\frac{1}{\sigma_k^2} \sum_{i \in [N]} r_{ik} (x_i - \mu_k) = 0 \\ \implies \mu_{k,\text{new}} &= \frac{\sum_{i \in [N]} r_{ik} x_i}{\sum_{i \in [N]} r_{ik}}.\end{aligned}$$

- ▶ Variance

$$\begin{aligned}\frac{\partial \mathcal{Q}}{\partial \sigma_k^2} &= \sum_{i \in [N]} r_{ik} \left[-\frac{D}{\sigma_k} + \frac{1}{\sigma_k^3} \|x_i - \mu_k\|^2 \right] = 0 \\ \implies \sigma_{k,\text{new}}^2 &= \frac{1}{D} \frac{\sum_{i \in [N]} r_{ik} \|x_i - \mu_{k,\text{new}}\|^2}{\sum_{i \in [N]} r_{ik}}\end{aligned}$$

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 $\theta = \{\pi_k, \mu_k, \sigma_k^2\}_{k \in [K]}$,

- ▶ 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:

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

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

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

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 :

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

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

- 1 Latent Variable Models
- 2 Example: Gaussian Mixture Model (GMM)
Expectation Maximization
- 3 Analysis and generalization of EM algorithm
Jensen's inequality and Gibb's inequality
Mean-Field Variational Approximation
Example with Logistic Regression
- 4 Variational Auto Encoders (VAEs)

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:

$$\begin{aligned} f\left(\sum_{i \in [N]} p_i x_i\right) &= f\left(p_1 x_1 + (1 - p_1) \left(\frac{\sum_{i=2}^N p_i x_i}{1 - p_1}\right)\right) \\ &\leq p_1 f(x_1) + (1 - p_1) f\left(\frac{\sum_{i=2}^N p_i x_i}{1 - p_1}\right) \\ &= p_1 f(x_1) + (1 - p_1) f\left(\frac{p_2}{1 - p_1} x_2 + \left(\frac{1 - \sum_{i=1}^2 p_i}{1 - p_1}\right) \left(\frac{\sum_{i=3}^N p_i x_i}{1 - \sum_{i=1}^2 p_i}\right)\right) \\ &\leq p_1 f(x_1) + (1 - p_1) \left(\left(\frac{p_2}{1 - p_1}\right) f(x_2) + \left(\frac{1 - \sum_{i=1}^2 p_i}{1 - p_1}\right) f\left(\frac{\sum_{i=3}^N p_i x_i}{1 - \sum_{i=1}^2 p_i}\right)\right) \\ &= p_1 f(x_1) + p_2 f(x_2) \left(1 - \sum_{i=1}^2 p_i\right) f\left(\frac{\sum_{i=3}^N p_i x_i}{1 - \sum_{i=1}^2 p_i}\right) \dots \end{aligned}$$

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 .

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 ², defined by:

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

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

²i.e., $q(x) = 0$ implies $p(x) = 0$

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 $\{p_i\}, \{q_i\}$.

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

Gibb's Inequality: Proof of the Equality

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

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

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 .

$\Rightarrow \sum_i p_i = 1 = \sum_i q_i \exp(\lambda - 1)$, it follows that $\lambda = 1$.

Hence, the minimal p should be identical to q , and $\text{KL}(p\|q) = 0$ on such choice of p .

A Lower Bound on the Log-Likelihood (1)

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

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

where we marginalize the latent variables z in the second equality.

For any distribution $q(z)$ of the latent variables z , we have

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

A Lower Bound on the Log-Likelihood (2)

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

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

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 $k = 1, 2, \dots$)

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

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

- ▶ 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.,

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

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

Monotonicity of EM Algorithm

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

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

which is zero only if $q(z) = p(z | x, \theta)$ (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 $k = 1, 2, \dots$)

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

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

- ▶ M-step: Find the parameters that maximize:

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

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

Exponential Family

The exponential family is a family probability distribution functions each of which has a special form given by

$$p(x | \theta) = h(x)g(\eta) \exp(\eta^\top u(x)) ,$$

where $\eta = \eta(\theta)$ is a function of θ , and $h(x)$, $u(x)$ and $g(\eta)$ are known. The function $g(\eta)$ normalizes the distribution so that

$$g(\eta) \int h(x) \exp(\eta^\top u(x)) dx = 1 .$$

where the integration is replaced with sum for the case of discrete x .

Example of Exponential Family: Bernoulli

Consider a Bernoulli variable x with mean $\theta \in (0, 1)$ of which distribution can be expressed as follows:

$$\begin{aligned} p(x | \theta) &= \text{Bern}(x | \theta) = \theta^x(1 - \theta)^{1-x} \\ &= \exp(x \log \theta + (1 - x) \log(1 - \theta)) \\ &= (1 - \theta) \exp\left(\log\left(\frac{\theta}{1 - \theta}\right)x\right), \end{aligned}$$

which implies that the Bernoulli variable is an exponential family with $\eta = \log\left(\frac{\theta}{1 - \theta}\right)$,

$$h(x) = 1, \quad u(x) = x, \quad \text{and} \quad g(\eta) = \frac{1}{1 + \exp(\eta)}.$$

(Note that η contains sufficient information for θ , i.e., η is **sufficient statistics** for θ)

EM for Exponential Family

Given a complete data $s = (x, z)$ modeled by a distribution of exponential family, we write the expected complete-data log-likelihood:

$$\begin{aligned}\mathcal{Q}(\theta; \theta^{(t)}) &:= \mathbb{E}_{z|x, \theta^{(k)}} [\log p(s | \theta)] \\ &= \mathbb{E}_{z|x, \theta^{(k)}} [\eta(\theta)^\top u(s)] + \mathbb{E}_{z|x, \theta^{(k)}} [\log(h(s))] + \log g(\eta(\theta)) \\ &= \eta(\theta)^\top \mathbb{E}_{z|x, \theta^{(k)}} [u(s)] + \mathbb{E}_{z|x, \theta^{(k)}} [\log(h(s))] + \log g(\eta(\theta)) .\end{aligned}$$

Hence, the EM algorithm is given as:

- ▶ E-step: $u^{(k+1)} = \mathbb{E}_{z|x, \theta^{(k)}} [u(s)]$.
- ▶ M-step: $\theta^{(k+1)} = \arg \max_{\theta} [\eta(\theta)^\top u^{(k+1)} + \log g(\eta(\theta))]$.

Variational Inference: Mean-Field Approximation

- ▶ We want to approximate the posterior $p(z | x)$ when direct computation is intractable.
- ▶ **Mean-field assumption:** Factorize the variational distribution into independent blocks:

$$q(z) = \prod_{j=1}^M q_j(z_j),$$

where $\{z_j\}$ are disjoint sets of the latent variables.

- ▶ The factorization is often called the *mean-field* assumption (each group of latent variables is treated as independent).
- ▶ This drastically reduces the complexity of inference by restricting how latent dimensions can interact.

Evidence Lower BOund (ELBO)

Starting from the log-marginal distribution

$$\begin{aligned}\log p(x) &= \log \int p(x, z) dz \\&= \log \int q(z) \frac{p(x, z)}{q(z)} dz. \\&\geq \int q(z) \log \left[\frac{p(x, z)}{q(z)} \right] dz \quad (\text{Jensen's ineq}) \\&= \underbrace{\mathbb{E}_{q(z)}[\log p(x, z)]}_{\text{(A) Expected log-likelihood}} - \underbrace{\mathbb{E}_{q(z)}[\log q(z)]}_{\text{(C) Negative entropy}} \\&= \underbrace{\mathbb{E}_{q(z)}[\log p(x | z)]}_{\text{(A) Expected log-likelihood}} + \underbrace{\mathbb{E}_{q(z)}[\log p(z)]}_{\text{(B) Log-prior}} - \underbrace{\mathbb{E}_{q(z)}[\log q(z)]}_{\text{(C) Negative entropy}}.\end{aligned}$$

Under mean-field, the optimization often splits into simpler subproblems.

ELBO and KL Divergence

Claim: Minimizing the KL divergence $\text{KL}(q(z) \parallel p(z | x))$ is equivalent to maximizing the *ELBO*:

$$\mathcal{L}(q) = \mathbb{E}_{q(z)} [\log p(x, z)] - \mathbb{E}_{q(z)} [\log q(z)].$$

The difference between the marginal and ELBO becomes

$$\log p(x) - \mathcal{L}(q) = \text{KL}(q(z) \parallel p(z | x)).$$

Thus,

$$\max_q \mathcal{L}(q) \iff \min_q \text{KL}(q(z) \parallel p(z | x)).$$

Coordinate Ascent for Mean-Field VI

Coordinate-Ascent Updates:

- ▶ Under the mean-field assumption $q(z) = \prod_j q_j(z_j)$, each factor $q_j(z_j)$ has an analytic update (in exponential-family models):

$$\log q_j(z_j) \leftarrow \mathbb{E}_{\substack{i \neq j \\ q_i(z_i)}} [\log p(x, z)] + \text{const.}$$

- ▶ We iterate over each factor $q_j(z_j)$, update it while holding the others fixed.
- ▶ This is repeated until the ELBO converges.

Advantages:

- ▶ Relatively straightforward to implement for many graphical models.
- ▶ Scales to large datasets if combined with stochastic optimizations.

Tradeoff:

- ▶ Mean-field underestimates posterior correlations (factorization).

Bayesian Logistic Regression: Model Setup

Data and Parameters:

- ▶ Observations: $\{(x_n, y_n)\}_{n=1}^N$, where $x_n \in \mathbb{R}^D$ and $y_n \in \{0, 1\}$.
- ▶ Unknown parameter: $w \in \mathbb{R}^D$.
- ▶ Prior on w : $p(w) = \mathcal{N}(w | 0, I)$.

Likelihood: For each n ,

$$p(y_n | x_n, w) = \sigma(w^\top x_n)^{y_n} \left[1 - \sigma(w^\top x_n)\right]^{1-y_n},$$

where $\sigma(u) = \frac{1}{1+e^{-u}}$ is the logistic function.

Posterior:

$$p(w | \{x_n, y_n\}) \propto p(w) \prod_{n=1}^N p(y_n | w, x_n).$$

Challenge: Intractable Posterior

- ▶ $\sigma(u)$ is non-conjugate with the Gaussian prior.
- ▶ \implies No closed-form expression for $p(w \mid \{x_n, y_n\})$.
- ▶ Classic solutions like *Laplace approximation* or *MCMC* can be more expensive.

Idea: Variational Inference (VI)

Approximate the true posterior with a simpler, tractable family $q(w)$ by *optimizing* a divergence measure (often KL).

Mean-Field Approximation

Factorized variational distribution:

$$q(w) = \prod_{j=1}^D q_j(w_j), \quad \text{with each } q_j(w_j) = \mathcal{N}(w_j \mid \mu_j, \sigma_j^2).$$

- ▶ Assumes $\{w_j\}$ are independent under q .
- ▶ Reduces complexity: no cross-covariances.
- ▶ Variational parameters: $\{\mu_j, \sigma_j^2\}_{j=1}^D$.

Tradeoff:

- ▶ *Pro:* Simpler updates, easier to scale.
- ▶ *Con:* Ignores correlations among components of w .

ELBO of Bayesian Logistic Regression

The ELBO³ objective of Bayesian Logistic Regression becomes

$$\begin{aligned}\mathcal{L}(q) &= \underbrace{\mathbb{E}_{q(w)}[\log p(\{y_n\} | w)]}_{\text{(A) Expected log-likelihood}} + \underbrace{\mathbb{E}_{q(w)}[\log p(w)]}_{\text{(B) Log-prior}} - \underbrace{\mathbb{E}_{q(w)}[\log q(w)]}_{\text{(C) Negative entropy}}. \\ &= \underbrace{\mathbb{E}_{q(w)}[\log p(\{y_n\} | w)]}_{\text{(A) Expected log-likelihood}} - \sum_j \underbrace{\text{KL}(q_j || p_j)}_{\text{KL between } q_j \text{ and } p_j}\end{aligned}$$

To update variational parameter μ_j , we need to compute $\frac{\partial \mathcal{L}}{\partial \mu_j}$.

First, it is known that $\frac{\partial \text{KL}(q_j || p_j)}{\partial \mu_j} = -\mu_j$

³We omit input x_n for compactness.

Partial derivative of (A) w.r.t. μ_j

Recall the Mean-Field Approximation:

$$q(w) = \prod_{d=1}^D \mathcal{N}(w_d | \mu_j, \sigma_j^2).$$

Hence, for each j ,

$$q_j(w_j) = \frac{1}{\sqrt{2\pi} \sigma_j} \exp\left(-\frac{(w_j - \mu_j)^2}{2 \sigma_j^2}\right).$$

Thus,

$$q(w) = \prod_{j=1}^D \left[\frac{1}{\sqrt{2\pi} \sigma_j} \right] \exp\left(-\frac{1}{2} \sum_{j=1}^D \frac{(w_j - \mu_j)^2}{\sigma_j^2}\right).$$

Target quantity:

$$\mathbb{E}_{q(w)}[\log p(\{y_n\} \mid w)] = \int_{\mathbb{R}^D} \left(\prod_{j=1}^D q_j(w_j) \right) \log p(\{y_n\} \mid w) \ dw_1 \dots dw_j.$$

We want

$$\frac{\partial}{\partial \mu_j} \mathbb{E}_{q(w)}[\log p(\{y_n\} \mid w)].$$

Two Common Ways to Derive It:

- 1. Score-Function Gradient.**
- 2. Reparameterization Trick.**

We cover the score-function gradient for now.

Score-Function Gradient

We write the derivative *inside* the integral:

$$\frac{\partial}{\partial \mu_j} \int q(w) \log p(\{y_n\} | w) dw = \int \frac{\partial}{\partial \mu_j} [q(w) \log p(\{y_n\} | w)] dw.$$

Since $\log p(\{y_n\} | w)$ does *not* directly depend on μ_j (only on w),

$$\frac{\partial}{\partial \mu_j} [q(w) \log p(\{y_n\} | w)] = \log p(\{y_n\} | w) \frac{\partial q(w)}{\partial \mu_j}.$$

Now note

$$\frac{\partial q(w)}{\partial \mu_j} = q(w) \frac{\partial}{\partial \mu_j} [\log q(w)].$$

$$\log q(w) = \sum_{j=1}^D \log q_j(w_j) = \sum_{j=1}^D \left[-\frac{1}{2} \log(2\pi\sigma_j^2) - \frac{(w_j - \mu_j)^2}{2\sigma_j^2} \right].$$

Hence,

$$\frac{\partial}{\partial \mu_j} \log q(w) = \frac{\partial}{\partial \mu_j} \left[-\frac{(w_j - \mu_j)^2}{2\sigma_j^2} \right] = \frac{(w_j - \mu_j)}{\sigma_j^2}.$$

Therefore,

$$\begin{aligned} \frac{\partial \mathbb{E}_{q(w)}[\log p(\{y_n\} | w)]}{\partial \mu_j} &= \int q(w) \log p(\{y_n\} | w) \frac{(w_j - \mu_j)}{\sigma_j^2} dw \\ &= \mathbb{E}_{q(w)} \left[\log p(\{y_n\} | w) \frac{(w_j - \mu_j)}{\sigma_j^2} \right]. \end{aligned}$$

\implies This is the “score-function” form. It is correct but can have high variance in practice.

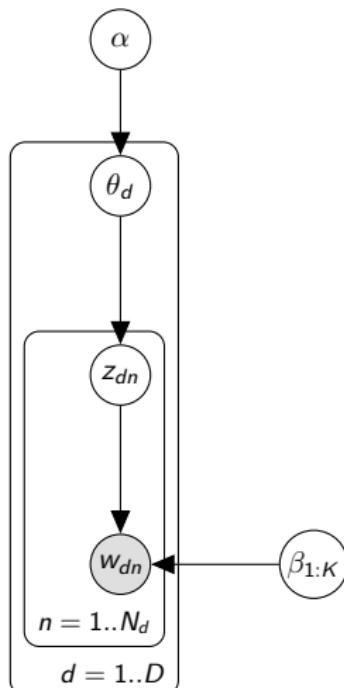
Combine all terms:

Putting all together, we get

$$\begin{aligned}\frac{\partial \mathcal{L}(q)}{\partial \mu_j} &= \frac{\partial \mathbb{E}_{q(w)}[\log p(\{y_n\} \mid w)]}{\partial \mu_j} - \frac{\partial \text{KL}(q_j \parallel p_j)}{\partial \mu_j} \\ &= \sum_{n=1}^N x_{n,j} \mathbb{E}_{q(w)}[y_n - \sigma(w^\top x_n)] - \mu_j. \\ &= \sum_{n=1}^N \mathbb{E}_{q(w)} \left[\log p(y_n \mid w) \frac{(w_j - \mu_j)}{\sigma_j^2} \right] - \mu_j.\end{aligned}$$

Similarly, we can compute $\frac{\partial \mathcal{L}(q)}{\partial \sigma_j}$ as well.

Example: Topic Model (Latent Dirichlet Allocation)



Generative Process (LDA):

1. $\beta_{1..K}$ are the topics, each β_k is a distribution over the vocabulary.
2. For each document d :
 - 2.1 Draw topic proportions $\theta_d \sim \text{Dirichlet}(\alpha)$.
 - 2.2 For each word $n \in \{1, \dots, N_d\}$:
 - 2.2.1 Draw topic assignment $z_{dn} \sim \text{Discrete}(\theta_d)$.
 - 2.2.2 Draw word $w_{dn} \sim \text{Discrete}(\beta_{z_{dn}})$.

Outline

- 1 Latent Variable Models
- 2 Example: Gaussian Mixture Model (GMM)
Expectation Maximization
- 3 Analysis and generalization of EM algorithm
Jensen's inequality and Gibb's inequality
Mean-Field Variational Approximation
Example with Logistic Regression
- 4 Variational Auto Encoders (VAEs)

Generative Models



Training data $\sim p_{\text{data}}(x)$



Generated samples $\sim p_{\text{model}}(x)$

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

- ▶ 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]

Manifold Hypothesis

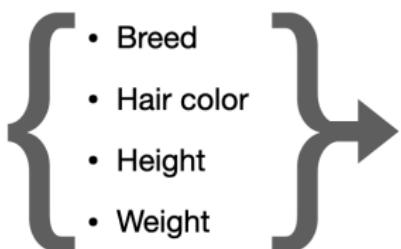
- ▶ To estimate the density of high-dimensional data, we assume each data point can be characterized by a few properties.
- ▶ For example, a cute cat can be (roughly) characterized by



- Breed
- Hair color
- Height
- Weight

Manifold Hypothesis

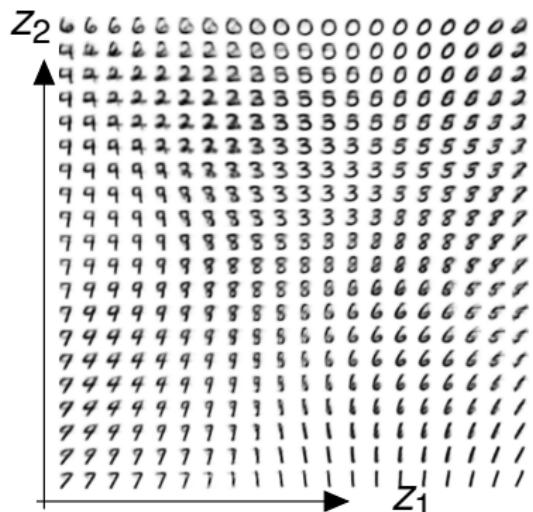
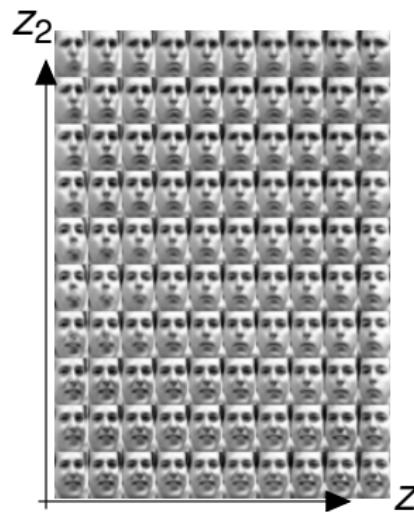
- ▶ To estimate the density of high-dimensional data, we assume each data point can be characterized by a few properties.
- ▶ For example, a cute cat can be (roughly) characterized by



Often unobservable!
Latent variables $\mathbf{z} \in \mathbb{R}^D$

Manifold Hypothesis

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



[Kingma and Welling 14]

A Probabilistic Approach for Generative Model

Given a single data point x , our objective is to maximize

$$p_{\theta}(x) = \int p(z)p_{\theta}(x | z)dz$$

where generative model is $p_{\theta}(x | z)$

- ▶ 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 $p_{\theta}(x | z)$ is anticipated to be complex, a neural network is widely selected
- ▶ The marginalization \int is intractable → variational inference

Intractability

- ▶ Data likelihood is intractable due to the integral:

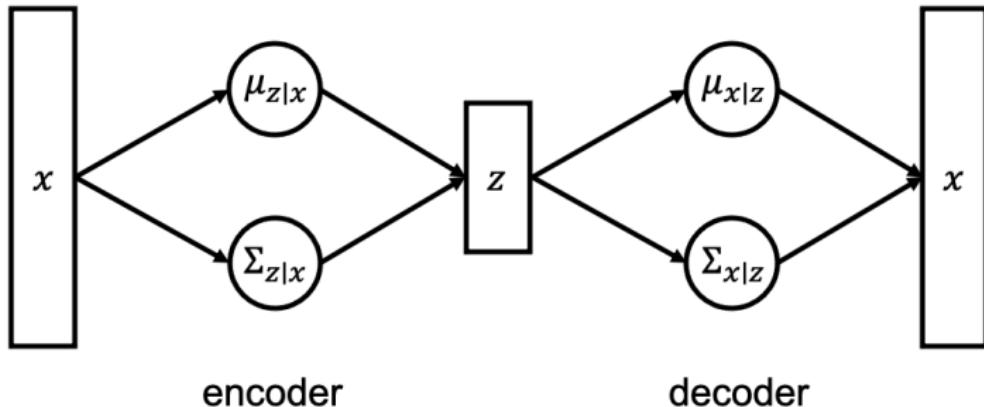
$$p_{\theta}(x) = \int p(z)p_{\theta}(x | z)dz$$

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

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

- ▶ A solution: approximate the posterior $p_{\theta}(z | x)$ using another (encoder) network $q_{\phi}(z | x)$
 - ▶ This can overcome the limitation of the mean-field approach.

A Probabilistic Framework of Auto-encoder



encoder

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

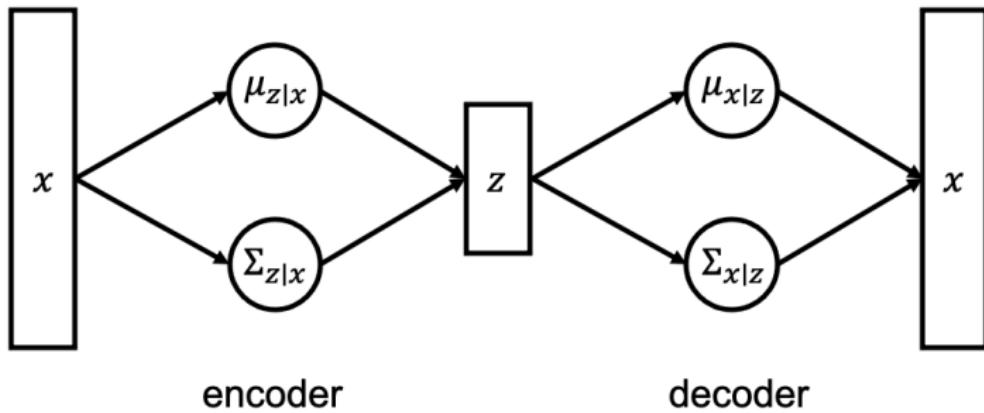
decoder

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

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

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

A Probabilistic Framework of Auto-encoder



$$q_\phi(z|x) = \mathcal{N}(\mu_{z|x}, \Sigma_{z|x}) \quad p_\theta(x|z) = \mathcal{N}(\mu_{x|z}, \Sigma_{x|z})$$

For decoder p_θ ,

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

Variational Autoencoder

Recalling we aim at MLE: for given x ⁴,

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

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

⁴ $p(x)p(z | x) = p(x | z)p(z)$

Variational Autoencoder

Recalling we aim at MLE: for given x ,

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

- ▶ Term (A) is tractable as we can sample $z \sim q_\phi(\cdot | x)$ from the encoder, and compute $p_\theta(x | z)$ 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 ($\text{KL} \geq 0$)
- ▶ Hence, define (A)+(B) as variational lower bound $\mathcal{L}(x, \theta, \phi)$
ELBO: Evidence Lower BOund and maximize it

Training VAE

Training VAE:

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

Understanding ELBO:

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

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

Training VAE: Monte Carlo Method

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

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

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:

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

We can approximate the expectation with Monte-Carlo method:

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

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

Training Decoder p

Given the Monte Carlo approximation

$$\begin{aligned}\mathbb{E}_z [\log p_\theta(x | z)] &\approx \frac{1}{N} \sum_{i=1}^N \log p_\theta(x | z^{(i)}) \\ &= -\log \sigma_x^2 \sqrt{2\pi} - \frac{1}{2N} \sum_{i=1}^N \frac{(x - g_\theta(z^{(i)}))^2}{\sigma_x^2}\end{aligned}$$

we can approximate the derivative w.r.t θ^5 .

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

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

⁵The same procedure can be applied if g_θ is a NN parameterized by θ .

Training Encoder q

Again, given the Monte Carlo approximation

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

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

Why? the distribution q is replaced by its samples!

⇒ 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

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

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

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

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

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

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

KL Divergence

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

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

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

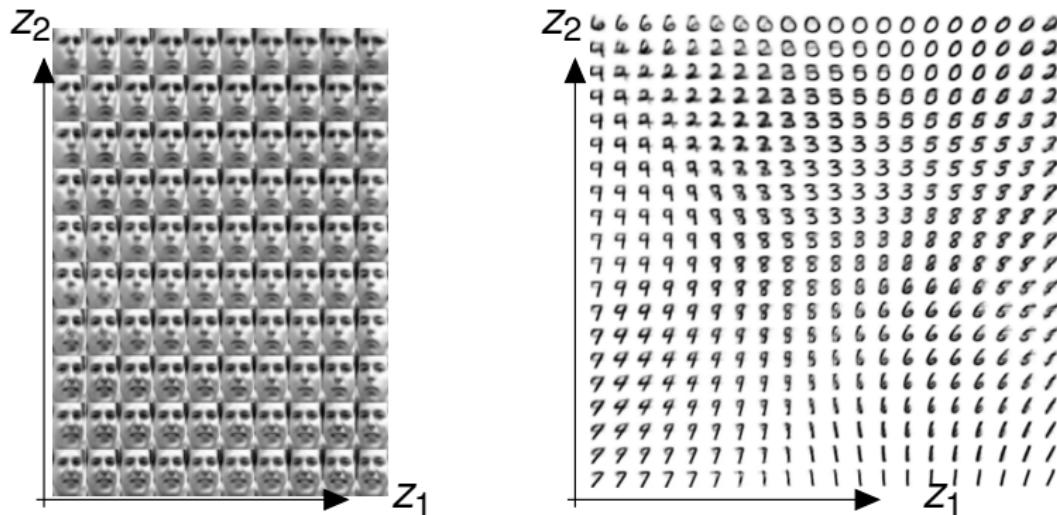
Training VAE via ELBO:

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

- ▶ $\partial \mathcal{L}(x, \theta, \phi) / \partial \theta$ 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

Summary

- ▶ Latent Variable Models (LVMs)
 - ▶ introduces causal relation between observed and unobserved variables
- ▶ Expectation-Maximization (EM) Algorithm
 - ▶ provides a practical solution to the intractable posterior
- ▶ Variational inference
 - ▶ generalized from the expectation maximization
- ▶ Variational Auto Encoders (VAEs)
 - ▶ incorporates the power of NNs in VI