Linear Regression & MLE

* Assumption : x, y in linear relationship (only 1 feature for each data x)

$$y = ax + b + \epsilon \qquad (\epsilon \sim N(0, \delta^2),\ \delta \text{ is constant})$$

(vector version for single data $\mathbf{x}$: $y = W^T\mathbf{x} + \epsilon$ where $\mathbf{x} = [1\ x_1\ x_2 \cdots x_n]$)

* Linear Regression Model

$$\hat{y} = Wx + b \quad \text{where } W, b \text{ are parameters}$$

(vector version : $\hat{y} = W^T\mathbf{x}$)

* MLE in Linear Regression? (For a single data y given x)

$\epsilon = y - \hat{y} \sim N(0, \delta^2)$ (where δ is constant)

$y \sim N(y; \hat{y}, \delta) \iff y \sim p(y|x; W, b)$ (δ is constant)

δ가 constant라서 parameter가 아님. 상수임.

$$\log P_\theta(y) = \log P(y|x; W, b) = \log N(y; \hat{y}, \delta)$$

$$= \log\left(\frac{1}{\sqrt{2\pi\delta^2}} e^{-\frac{(y-\hat{y})^2}{2\delta^2}}\right)$$

$$= \log\frac{1}{\sqrt{2\pi\delta^2}} + \log e^{-\frac{(y-\hat{y})^2}{2\delta^2}}$$

$$= -\frac{1}{2\delta^2}(y-\hat{y})^2 + \log\frac{1}{\sqrt{2\pi\delta^2}}$$

$$= -\frac{1}{2\delta^2}(y - Wx - b)^2 + \log\frac{1}{\sqrt{2\pi\delta^2}}$$

does not change (since parameters are only W, b)

$$\underset{W,b}{\operatorname{argmax}} \log N(y; \hat{y}, s)$$

$$\Leftrightarrow \underset{W,b}{\operatorname{argmax}} \log P(y|x; W, b)$$

$$\Leftrightarrow \underset{W,b}{\operatorname{argmax}} -\frac{1}{2s^2}(y - Wx - b)^2$$

$$\Leftrightarrow \underset{W,b}{\operatorname{argmax}} -(y - Wx - b)^2$$

$$\Leftrightarrow \underset{W,b}{\operatorname{argmin}} (y - Wx - b)^2$$

$$\Leftrightarrow \underset{W,b}{\operatorname{argmin}} (y - (Wx+b))^2 \quad \text{(square of residual!)}$$

⇝ This is why objective function for Linear Regression leads to MSE Loss.

For n datas?

$$\underbrace{L(W,b)}_{\text{object function}} = \frac{1}{N}\sum_{n=1}^{N} (y^{(n)} - (Wx^{(n)}+b))^2 \quad \text{(MSE)}$$

VAE : Variational Autoencoder

→ 학습하는 data에 대한 확률적 그 분포에서 확률분포 ($P_\theta(x)$를 신경망으로 구현)

↳ purpose of generative models

* Review

- Single normal distribution (Multivariate)

$P_\theta(x) = N(x;\theta)$ where $\theta = \{u, \Sigma\}$

MLE for D

$D = \{x^{(1)}, x^{(2)} \cdots x^{(N)}\}$

$L(D;\theta)$

$= \log P_\theta(D)$

$= \log\left(\prod_{n=1}^{N} P_\theta(x^{(n)})\right)$

$= \sum_{n=1}^{N} \log P_\theta(x^{(n)})$

Solve : $\frac{\partial}{\partial\theta} L(D;\theta) = 0$

- GMM : Gaussian Mixture Model

$\frac{\partial}{\partial\theta} \log_\theta(D) = 0$ → 해석적으로 풀 수 X ($p(x,z)$에서 $p(x)$를 도출하기 위해서는 $\sum_z$ marginalization이 필요하고 이는 log-sum을 유발)

use ELBO,

$ELBO(D; q, \theta) \le \log P_\theta(D)$

$\iff \sum_{n=1}^{N}\sum_{z^{(n)}} q^{(n)}(z^{(n)}) \log \frac{P_\theta(x^{(n)}, z^{(n)})}{q^{(n)}(z^{(n)})} \le \log P_\theta(D)$

E-M with ELBO (θ, q update stepwisely)

E: $q^{(n)}(z^{(n)}) = P_\theta(z^{(n)}|x^{(n)})$ (for all n datas & for all K latent variables), ELBO log-likelihood approx.

M: $\frac{\partial}{\partial\theta} ELBO = 0$ → maximize

latent variable (z) generation model

✳ Sampling (or generating) data with VAE (Decoder)

Parameter is fixed!

① Sample latent variable z from "fixed" normal distribution $(P(z) = N(z; 0, I))$

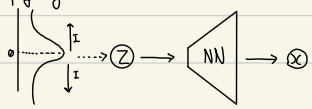
② Transform latent variable z to observed data x using neural network (Decoder)

zero vector, Identity Matrix

where $z \in D^H$, fixed params are $\{0, I\}$

✳ ···> : sampling

· Sampling figure



① $\therefore p(z) = N(z; 0, I)$

- In GMM z follows categorical distribution where z is discrete variable. In VAE, z is sampled from gaussian distribution $(N(z; 0, I))$ where z is continuous variable which makes representation broader.

↳ 더 다양하고 풍부한 데이 표현 가능

population

- VAE is also a generative model which estimates the observed variable x's distribution $P(x)$ based on sample.

Since VAE transforms z to x, this models probability $P(x|z)$

Since VAE decoder outputs $\hat{x}$ from sampled z, define the x's distribution as..

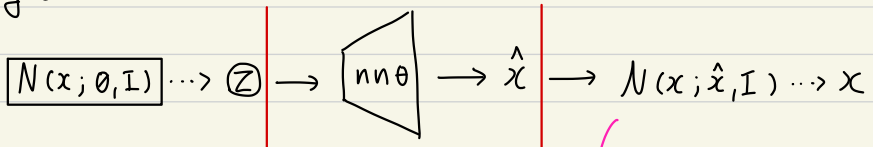
vector

② $\hat{x} = \text{NeuralNet}(z; \theta)$

Identity matrix

③ $P_\theta(x|z) = N(x; \hat{x}, I) \quad (x \sim N(x; \hat{x}, I))$

✳ figure



If the observed data is binary $\{0, 1\}$ $P_\theta(x|z)$ can be modeled as Bernoulli dist.

✳ Limitation of EM in VAE (Difficulty in E-step)

- ELBO EM Estep for n datas.

  renew $q^{(n)}(z^{(n)}) = P_\theta(z^{(n)} | x^{(n)})$

Review) latent variable model $p(x,z)$ 에서 x 의 대한 주변 $p(x) = \int p(x,z)\,dz$ 이다.

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

for VAE, $P_\theta(z^{(n)} | x^{(n)}) = \frac{P_\theta(x^{(n)}, z^{(n)})}{P_\theta(x^{(n)})}$

$$= \frac{P_\theta(x^{(n)}, z^{(n)})}{\int P_\theta(x^{(n)}, z^{(n)})\,dz}$$ → marginalization

$\int P_\theta(x, z)\,dz$ is easily countable in GMM since latent variable z is discrete.

$\Leftrightarrow \sum_z P_\theta(x, z)\,dz$

But in VAE z is continuous and z is vector → $\int P_\theta(x,z)\,dz$ is impossible (or very hard)

∴ VAE 에서는 EM의 E-step을 direct하게 적용할 수 X. → Needs improvement in EM algorithm.

"How To Solve?"

✳ VAE training by improving EM algorithm (Encoder)

- Review : Derive ELBO from $\log P_\theta(x)$ using $q_\phi(z)$!

$\log P_\theta(x)$ (log likelihood for single data x)

$= \int q_\phi(z)\,dz \log P_\theta(x) \quad \left(\int q_\phi(z)\,dz = 1\right)$

$= \int q_\phi(z) \log P_\theta(x)\,dz$

$= \int q_\phi(z) \log \frac{P_\theta(x,z)}{P_\theta(z|x)}\,dz$

$= \int q_\phi(z) \log \frac{P_\theta(x,z)}{P_\theta(z|x)} \frac{q_\phi(z)}{q_\phi(z)}\,dz \quad \left(\frac{q_\phi(z)}{q_\phi(z)} = 1\right)$

$= \int q_\phi(z) \left(\log \frac{P_\theta(x,z)}{q_\phi(z)} + \log \frac{q_\phi(z)}{P_\theta(z|x)}\right) dz$

$= \int q_\phi(z) \log \frac{P_\theta(x,z)}{q_\phi(z)}\,dz + \int q_\phi(z) \log \frac{q_\phi(z)}{P_\theta(z|x)}\,dz$

$\text{ELBO}(x; \phi, \theta)$ $\qquad$ $D_{KL}(q_\phi(z) \,||\, P_\theta(z|x))$

$\therefore \log P_\theta(x) = \int q_\phi(z) \log \frac{P_\theta(x,z)}{q_\phi(z)}\,dz + D_{KL}(q_\phi(z) \,||\, P_\theta(z|x)) \quad (D_{KL} \geq 0)$

$\geq \int q_\phi(z) \log \frac{P_\theta(x,z)}{q_\phi(z)}\,dz \quad (\text{ELBO})$

According to EM algorithm, E fixes θ and renews $q_\phi(z) = P_\theta(z|x)$, but according to previous page it is hard to get $P_\theta(z|x)$ since z is continual vector.

✳ $P(z)$ VS $q_\phi(z)\,(= q_\phi(z|x))$

for generative model $p(x,z)$ (with latent variable)

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

Since Ⓐ is not easily calculable and optimize (for D)

we use $q_\phi(z|x)$ (or $q_\phi(z)$) to approximate $p(z|x)$

· Comparison

$p(z)$	$p(z|x)$
사전 분포 (Prior)	근사 분포 (Variational Posterior)
모델이 가정하는 z에 대한 분포	데이터 x에 대해 최적화된 분포
z에 대한 가정 (Generation시)	

★ $P(z)$ VS $q(z)\ (=q(z|x))$

for generative model $p(x,z)$ (with latent variable)

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

Since Ⓐ is not easily calculable and optimize (for D)

we use $q(z|x)$ (or $q(z)$) to approximate $p(z|x)$

Comparison?

$p(z)$

- 사전 분포 (Prior)
- 모델이 가정하는 z에 대한 분포
- z에 대한 가정 / Generation시 이용되는 z 분포
  - 모델이 정의
- $\log P(x) = \log \int p(z)p(x|z)\,dz$ 가정

$p(z|x)$

- 근사 분포 (Variational Posterior)
- 데이터 x에 대해 조건부로 정의된 분포
- ELBO 계산을 위한 계산 가능한 분포, $p(z|x)$를 근사하기 위해
- 사용자 도입
- $-ELBO = E_{q(z|x)}[\log p(x|z)] - D_{KL}[q(z|x)\,||\,p(z)]$를 가정
- VAE에서 Encoder에 의해 구현됨

✳ Solution

① $q(z)$를 정규분포로 제한, 제한한 $q(z)$의 매개변수 $\psi = \{\mu, \Sigma\} \Leftrightarrow q_\psi(z) = N(z; \mu, \Sigma)$

② $q(z)$를 정규분포로 제한한 상태에서 ELBO를 최대화한다.

$$\log P_\theta(x) = \underbrace{\int q_\psi(z) \log \frac{P_\theta(x,z)}{q_\psi(z)} dz}_{\text{ELBO}} + D_{KL}(q_\psi(z) \,||\, p_\theta(z|x)) \quad \cdots ①$$

$$\theta, \psi = \underset{\theta,\psi}{\text{argmax}}\ \text{ELBO}(x; \theta, \psi) = \underset{\theta,\psi}{\text{argmax}} \int q_\psi(z) \log \frac{P_\theta(x,z)}{q_\psi(z)} dz$$

In VAE, cannot find $P_\theta(z|x)$ for $q_\psi(z) = P_\theta(z|x)$ with ψ

But can fit $q_\psi(z) = P_\theta(z|x)$ by maximizing ELBO (명시적으로 E step을 수행하지 않아도, ELBO를 maximize 하는 과정에서 $q_\psi(z)$가 $P_\theta(z|x)$에 fit된다)

Why? ①에서 좌항 $\log P_\theta(x)$는 ψ를 가지고 있지 않음

∴ ψ가 변해도 ELBO와 KL의 총합인 $\log P_\theta(x)$는 변하지 X

$\Leftrightarrow$ ψ에 대해 ELBO가 최대가 되면, KL항은 최소가 됨.

$\Leftrightarrow$ $q_\psi(z) \approx P_\theta(z|x)$에 가까워짐. (E step 수행가능)

(간단한 확률분포 $q_\psi(z)$를 이용해 계산이 불가능한 $P_\theta(z|x)$를 근사시킴. 이를 variational approximation 또는 variational bayes 라고함.)

변분 근사

* For n datas?

$$\sum_{n=1}^{N} ELBO(x^{(n)}; \theta, \psi^{(n)}) = \sum_{n=1}^{N} \int q_{\psi^{(n)}}(z) \log \frac{p_\theta(x^{(n)}, z)}{q_{\psi^{(n)}}(z)} dz$$

- prepare $q_{\psi^{(n)}}(z)$ for each $x^{(n)}$ where $\psi^{(n)} = \{\mu^{(n)}, \Sigma^{(n)}\}$ ... ①
- $\underset{\psi^{(n)}}{\text{argmax}} ELBO(x^{(n)}; \theta, \psi^{(n)}) \Longleftrightarrow q_{\psi^{(n)}}(z) \approx p_\theta(z|x^{(n)})$ (각각 $\psi^{(n)}$에 대해 ELBO 최대화)

"Prepare $\psi^{(n)}$ for each data $x^{(n)}$? what if n=1 billion?"

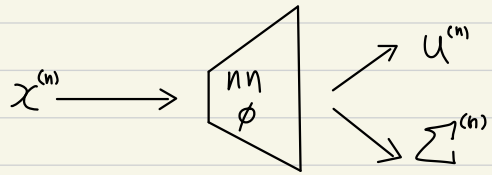
↓

"Use Neural Network for extracting $\psi^{(n)} = \{\mu^{(n)}, \Sigma^{(n)}\}$ for each $x^{(n)}$"

↓

"The role of Encoder in VAE"

* Encoder in VAE



- 근사사후 분포 $q(z)$의 매개변수를 nn으로 출력하는 기법을 amortized inference라고 함. (분할 상환 추론)

$z^{(n)} \in D^H \longrightarrow u^{(n)} \in D^H,\ \Sigma^{(n)} \in D^{H \times H}$

Refine Σ to diagonal matrix

$\Leftrightarrow \Sigma = \begin{bmatrix} \sigma_1^2 & & 0 \\ & \sigma_2^2 & \\ & & \ddots \\ 0 & & & \sigma_H^2 \end{bmatrix}$

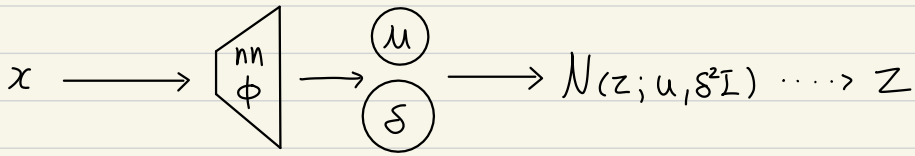
$\Leftrightarrow \Sigma$ is expressible as $\sigma^2 I$ where $\sigma \in D^H$ (elements are standard deviation of each feature), $I = I_H$

$\therefore\ u, \sigma = \text{NeuralNetEncoder}(x; \phi)$

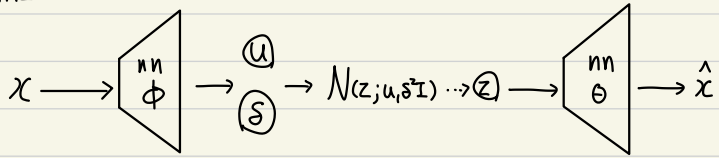
$q_\phi(z|x) = N(z; u, \sigma^2 I)$

※ EM에서는 $x^{(n)}$에 대한 $q^{(n)}(z)$이고, VAE에서는 $q^{(n)}(z)$의 parameter를 x로부터 신경망을 통해 얻어내기에, $q_\phi(z|x)$로 표현 (파라미터 ϕ로 학습된)

- Encoder



- VAE



* ELBO Optimization in VAE

- Decoder

$$p(z) = N(z; 0, I)$$
$$\hat{x} = \text{Neural Net}(z; \theta)$$
$$p_\theta(x|z) = N(x; \hat{x}, I)$$ where θ is decoder parameter

- Encoder

$$u, \delta = \text{Neural Net}(x; \phi)$$
$$q_\phi(z|x) = N(z; u, \delta^2 I)$$ where ϕ is encoder parameter

- ELBO

Single Sample x : $ELBO(x; \theta, \phi) = \int q_\phi(z|x) \log \frac{p_\theta(x,z)}{q_\phi(z|x)} dz$

$$= \int q_\phi(z|x) \log \frac{p_\theta(x|z)p(z)}{q_\phi(z|x)} dz$$

$$= \int q_\phi(z|x) \log p_\theta(x|z) dz + \int q_\phi(z|x) \log \frac{p(z)}{q_\phi(z|x)} dz$$

$$= \int q_\phi(z|x) \log p_\theta(x|z) dz - \int q_\phi(z|x) \log \frac{q_\phi(z|x)}{p(z)} dz$$

$$= \underbrace{E_{q_\phi(z|x)}[\log p_\theta(x|z)]}_{J_1} - \underbrace{D_{KL}(q_\phi(z|x) \,||\, p(z))}_{J_2}$$

$D = \{x^{(1)}, x^{(2)} \cdots x^{(N)}\}$: $ELBO(D; \theta, \phi) = \sum_{n=1}^{N} ELBO(x^{(n)}; \theta, \phi) = \sum_{n=1}^{N} \int q_\phi(z|x) \log \frac{p_\theta(x,z)}{q_\phi(z|x)} dz$

$$ELBO(x;\theta,\phi) = \underbrace{E_{q_\phi(z|x)}[\log P_\theta(x|z)]}_{J_1} - \underbrace{D_{KL}(q_\phi(z|x)\,||\,p(z))}_{J_2} \rightarrow \text{J_1은 최대화 J_2는 최소화 해야함 (ϕ,θ에 대해)}$$

- J_1

J_1은 $q_\phi(z|x)$를 따르는 z에 대한 $\log P_\theta(x|z)$의 Expectation

Monte carlo approximation을 통해 $q_\phi(z|x)$에서 z를 "1"개만 Sampling해서 근사

$$\mu, \sigma = \text{NeuralNet}(x;\phi)$$
$$z \sim N(z;\mu,\sigma^2 I) \quad \text{(sample only 1)}$$
$$\hat{x} = \text{NeuralNet}(z;\theta)$$
$$J_1 \approx \log P_\theta(x|z)$$

$$\Leftrightarrow J_1 \approx \log N(x;\hat{x},I)$$

$$= \log\left(\frac{1}{\sqrt{(2\pi)^D|I|}}\exp\left(-\frac{1}{2}(x-\hat{x})^T I^{-1}(x-\hat{x})\right)\right)$$

$$= -\frac{1}{2}(x-\hat{x})^T(x-\hat{x}) + \log\frac{1}{\sqrt{(2\pi)^D}} \quad (I^{-1}=I,\ |I|=1)$$

$$= -\frac{1}{2}\sum_{d=1}^{D}(x_d-\hat{x}_d)^2 + \underbrace{\log\frac{1}{\sqrt{(2\pi)^D}}}_{\text{constant}}$$

($\hat{x}$: defined by θ,ϕ)

$$\arg\max_{\theta,\phi} J_1 = \arg\max_{\theta,\phi} -\frac{1}{2}\sum_{d=1}^{D}(x_d-\hat{x}_d)^2$$

$$= \arg\min_{\theta,\phi} \sum_{d=1}^{D}(x_d-\hat{x}_d)^2 \quad \text{: reconstruction error}$$

· J_2

- Minimize J_2 for maximizing ELBO $(x; \phi, \theta)$

$q_\phi(z|x) : N(z; u, \sigma^2 I)$

$p(z) : N(z; 0, I)$

why Ⓐ? : $D_{KL}(q_\phi(z|x) \| p(z))$ 값을 해석적으로 계산 위함.

$q_\phi(z) = N(z; u_1, \sigma_1^2 I)$, $p(z) = N(z; u_2, \sigma_2^2 I)$ (두 확률분포가 Normal distribution일 때)

$D_{KL}(q\|p) = -\frac{1}{2}\sum_{h=1}^{H}\left(1+\log\frac{\sigma_{1,h}^2}{\sigma_{2,h}^2} - \frac{(u_{1,h}-u_{2,h})^2}{\sigma_{2,h}^2} - \frac{\sigma_{1,h}^2}{\sigma_{2,h}^2}\right)$ ($D_{KL}(p\|q)$도 해석적으로 구할 수 있음) ··· ①

$J_2 = D_{KL}(q_\phi(z|x) \| p(z))$

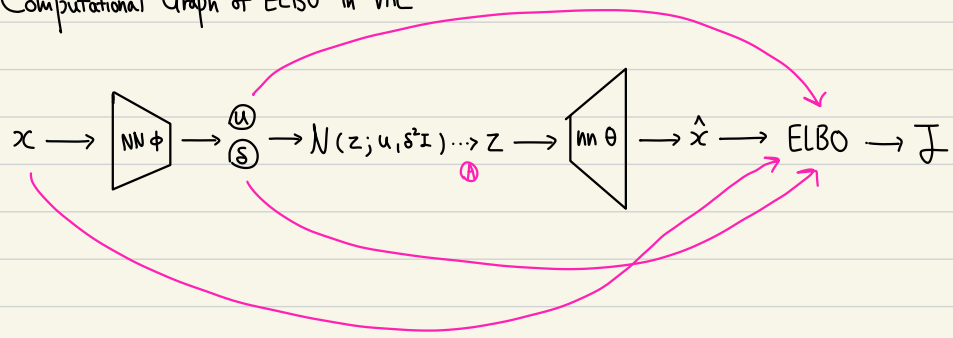
$= -\frac{1}{2}\sum_{h=1}^{H}(1+\log\sigma_h^2 - u_h^2 - \sigma_h^2)$

Minimize J_2 ? $\Leftrightarrow$ $q_\phi(z|x) = p(z)$ (consistency / regularization term)

$\therefore ELBO(x, \theta, \phi) \approx -\frac{1}{2}\sum_{d=1}^{D}(x_d - \hat{x}_d)^2 + \frac{1}{2}\sum_{h=1}^{H}(1+\log\sigma_h^2 - u_h^2 - \sigma_h^2) + const$

reconstruction loss / regularization term

- Computational Graph of ELBO in VAE



- update θ, ϕ simultaneously with G.D for maximizing ELBO

↳ 학습 두 단계로 나누어 θ, ϕ를 차례로 갱신함 (EM algorithm)

- issue : Gradient flow is unavailabe in Ⓐ

* Reparameterization trick for solving issue

- Make sampling to enable gradient flow

$z \sim N(z; u, \delta^2 I)$ sampling $\Leftrightarrow$ $\epsilon \sim N(\epsilon; 0, I)$

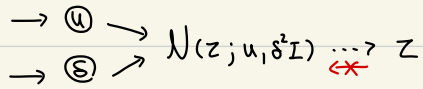
$z = u + \delta \odot \epsilon$ where $\odot$ is elementwise multiplication

$$z = u + \delta \odot \epsilon$$

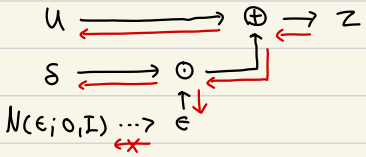
$$= \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_H \end{bmatrix} + \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_H \end{bmatrix} \odot \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_H \end{bmatrix}$$

$$= \begin{bmatrix} u_1 + \delta_1 \epsilon_1 \\ u_2 + \delta_2 \epsilon_2 \\ \vdots \\ u_H + \delta_H \epsilon_H \end{bmatrix}$$

- Without reparameterization trick



- With reparameterization trick



"Gradient flow available for Encoder"