

딥러닝 에스프레소

딥러닝을 위한 베이지안 통계 Day2

K-means, GMM, PCA, PPCA, Auto-Encoder

2020
멀티캠퍼스

박수철

K-means

K-means

- Clustering은 unsupervised learning으로 target value가 없는 dataset이 주어집니다.
- K-means는 machine learning의 가장 기초적인 clustering 알고리즘입니다.
- K개의 중심(centroid)을 찾고, 데이터들은 가장 가까운 중심에 해당하는 cluster에 대응시킵니다.
- K개의 centroid들은 각 cluster들의 평균 지점(mean)에 해당하기 때문에 K-means라는 이름이 붙었습니다.
- Dataset $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ 에 대해 K 개로 clustering한다고 합시다. 각 cluster의 mean이 μ_k 이고 data point \mathbf{x}_n 이 k 번째 cluster에 속함을 indicator variable $r_{nk} \in \{0,1\}$ 로 표현하여 다음과 같이 objective function을 쓸 수 있습니다.

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \| \mathbf{x}_n - \boldsymbol{\mu}_k \|^2$$

K-means

- Objective function J 를 최소화하기 위한 r_{nk} 는 각 data point index n 에 대해 $\| \mathbf{x}_n - \boldsymbol{\mu}_k \| ^2$ 항이 최소가 되는 cluster index k 를 대응시키는 방법으로 구할 수 있습니다.

●

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg \min_j \| \mathbf{x}_n - \boldsymbol{\mu}_k \| ^2 \\ 0 & \text{otherwise} \end{cases}$$

- data point \mathbf{x}_n 에 대응하는 cluster를 추정한다, 예상한다라는 맥락에서 위 과정을 expectation step이라 부릅니다.

- Objective function J 를 최소화하는 $\boldsymbol{\mu}_k$ 는 J 를 미분한 식을 0으로 두고 $\boldsymbol{\mu}_k$ 에 대해 식을 전개하여 얻을 수 있습니다.

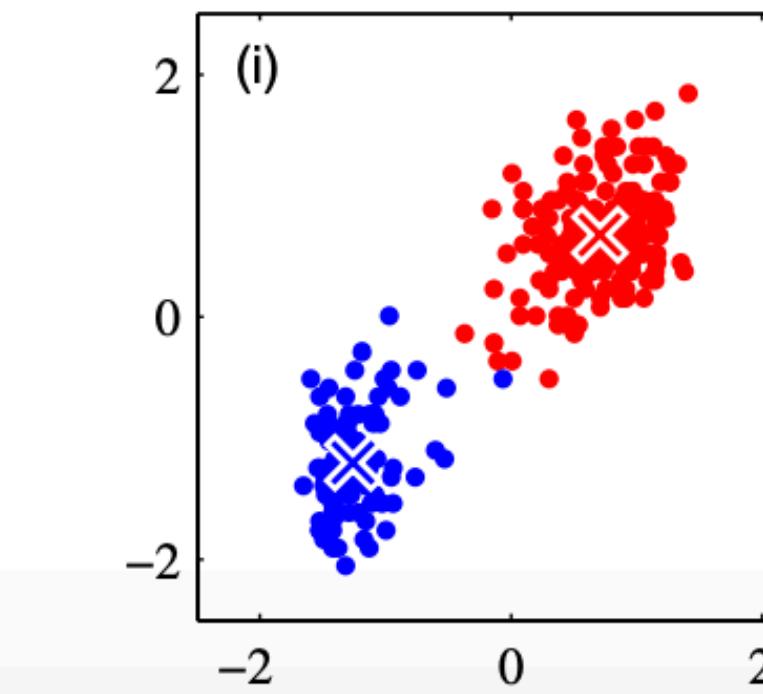
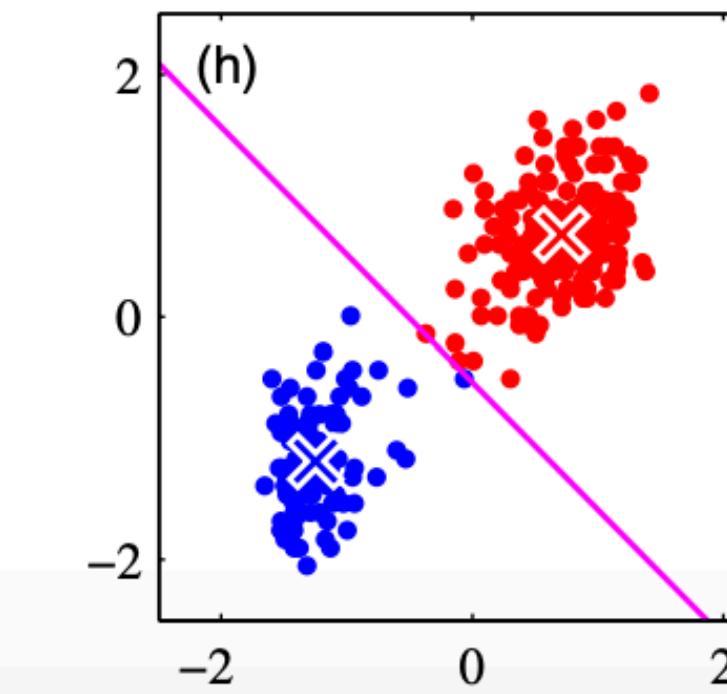
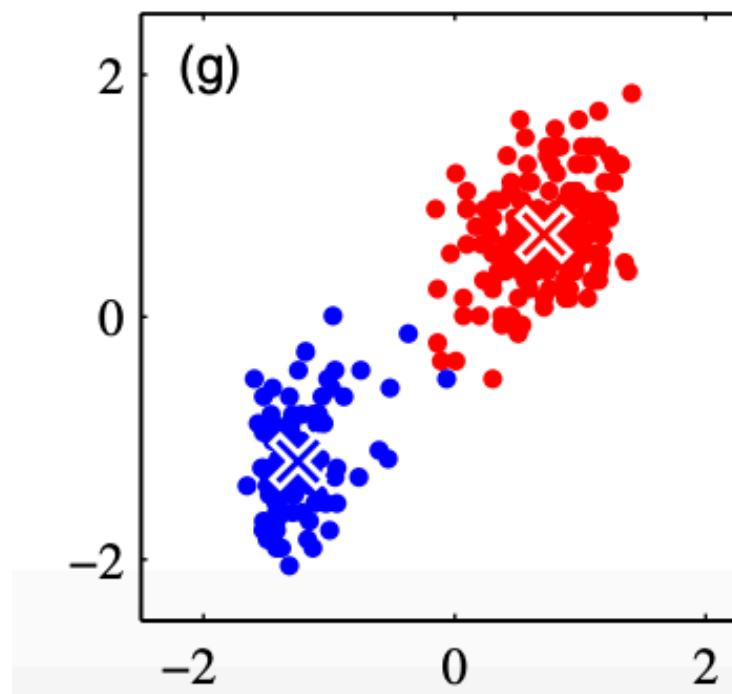
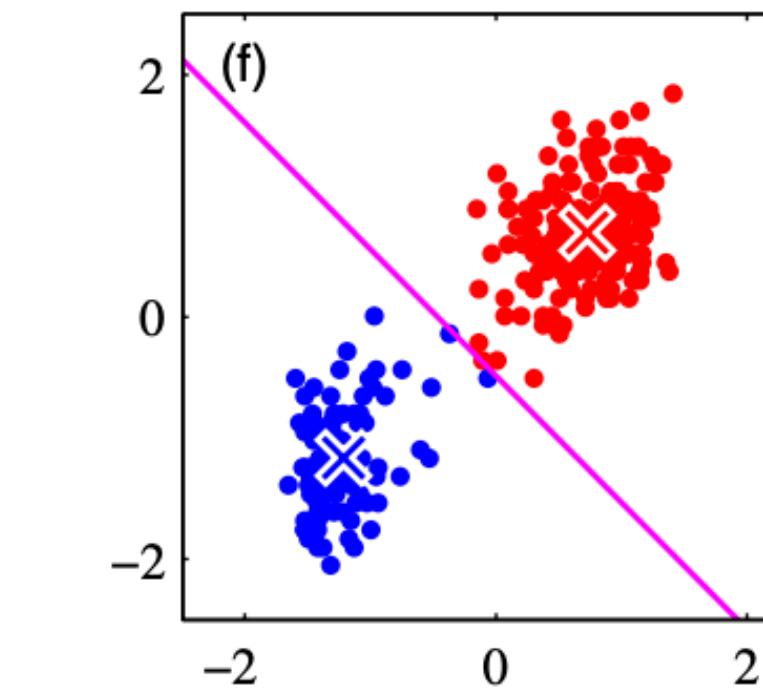
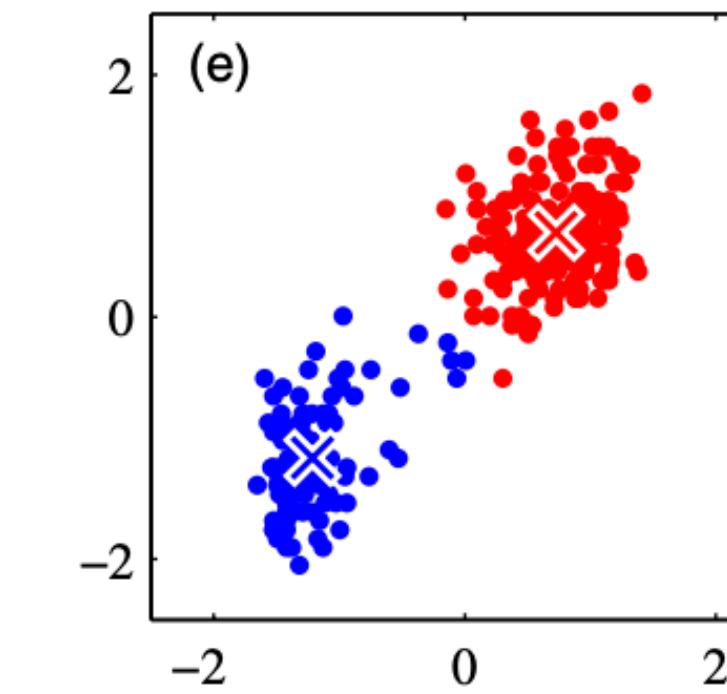
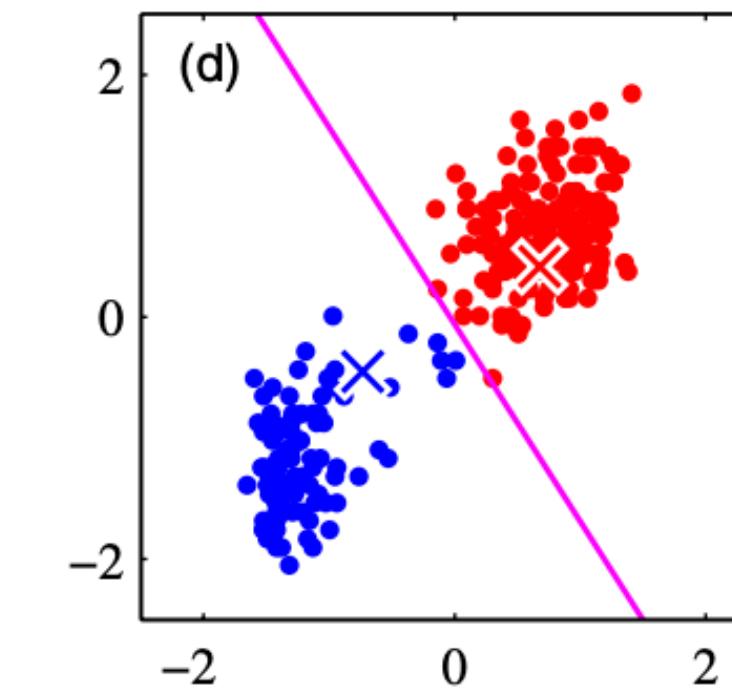
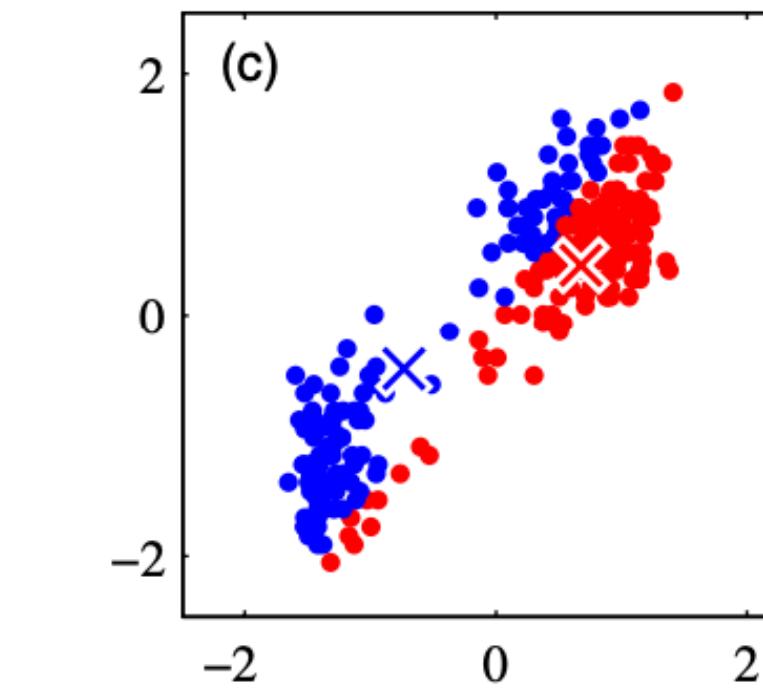
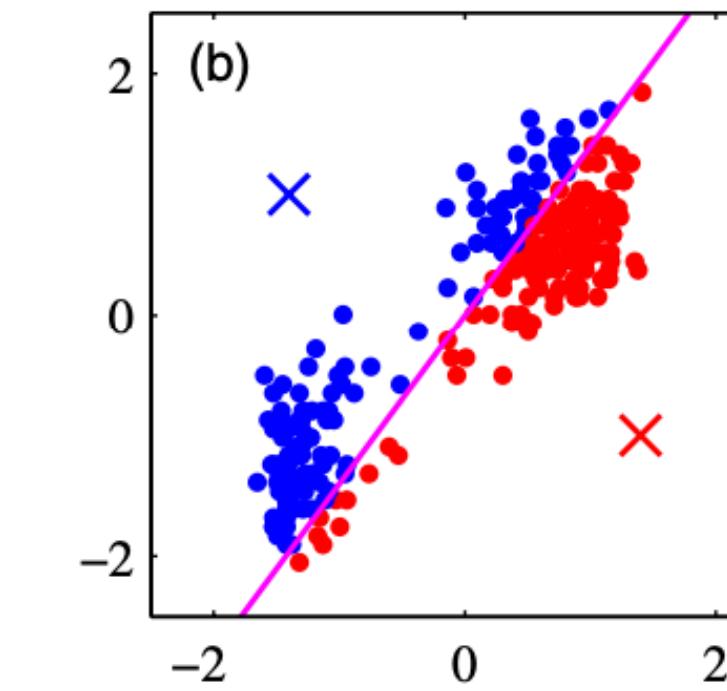
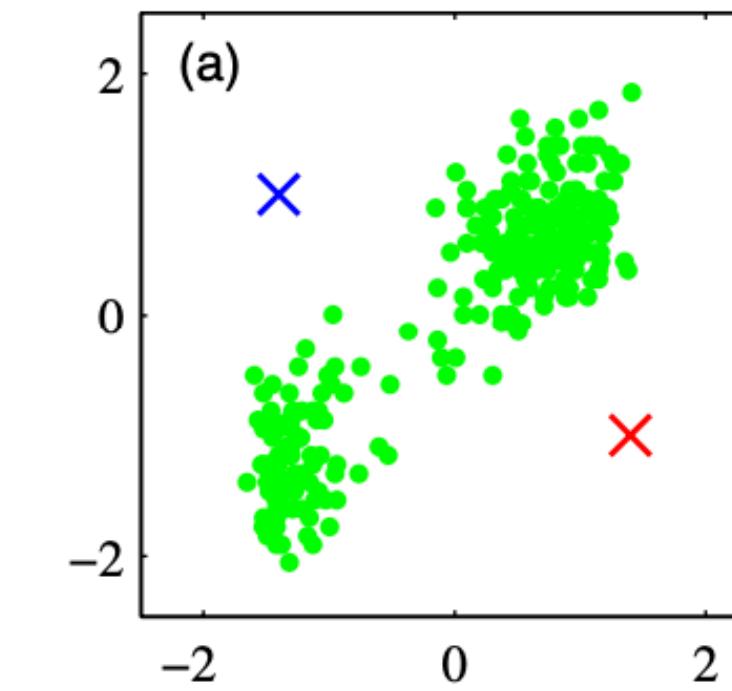
●

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}}$$

- Expectation step에서 추정한 r_{nk} 값들을 근거로 J 를 최대화하기 위한 parameter $\boldsymbol{\mu}_k$ 를 구한다는 맥락에서 위 과정을 maximization step이라고 부릅니다.

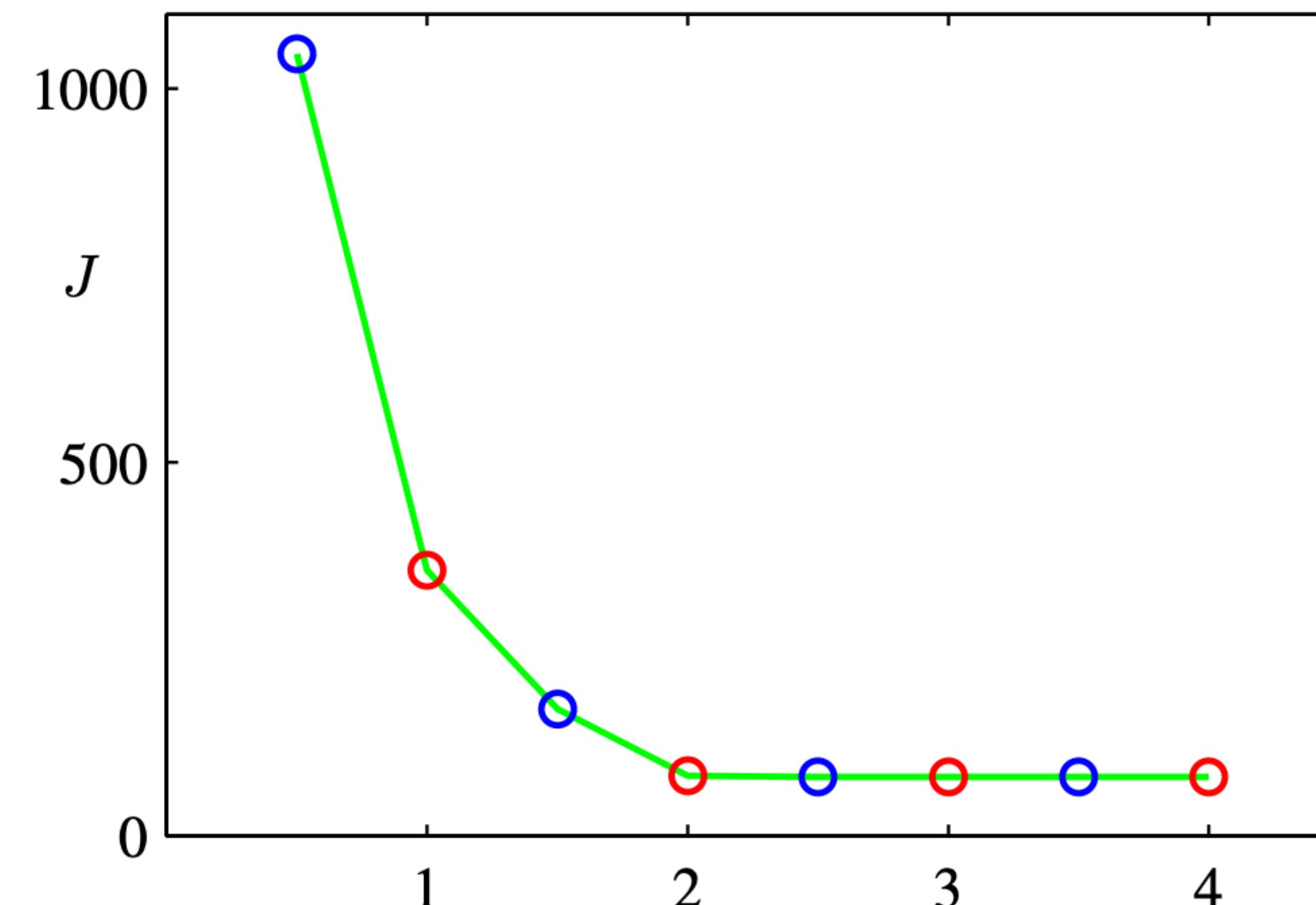
- Expectation과 maximization step을 번갈아 가면서 작동시키는 것을 EM 알고리즘이라 부릅니다.

K-means



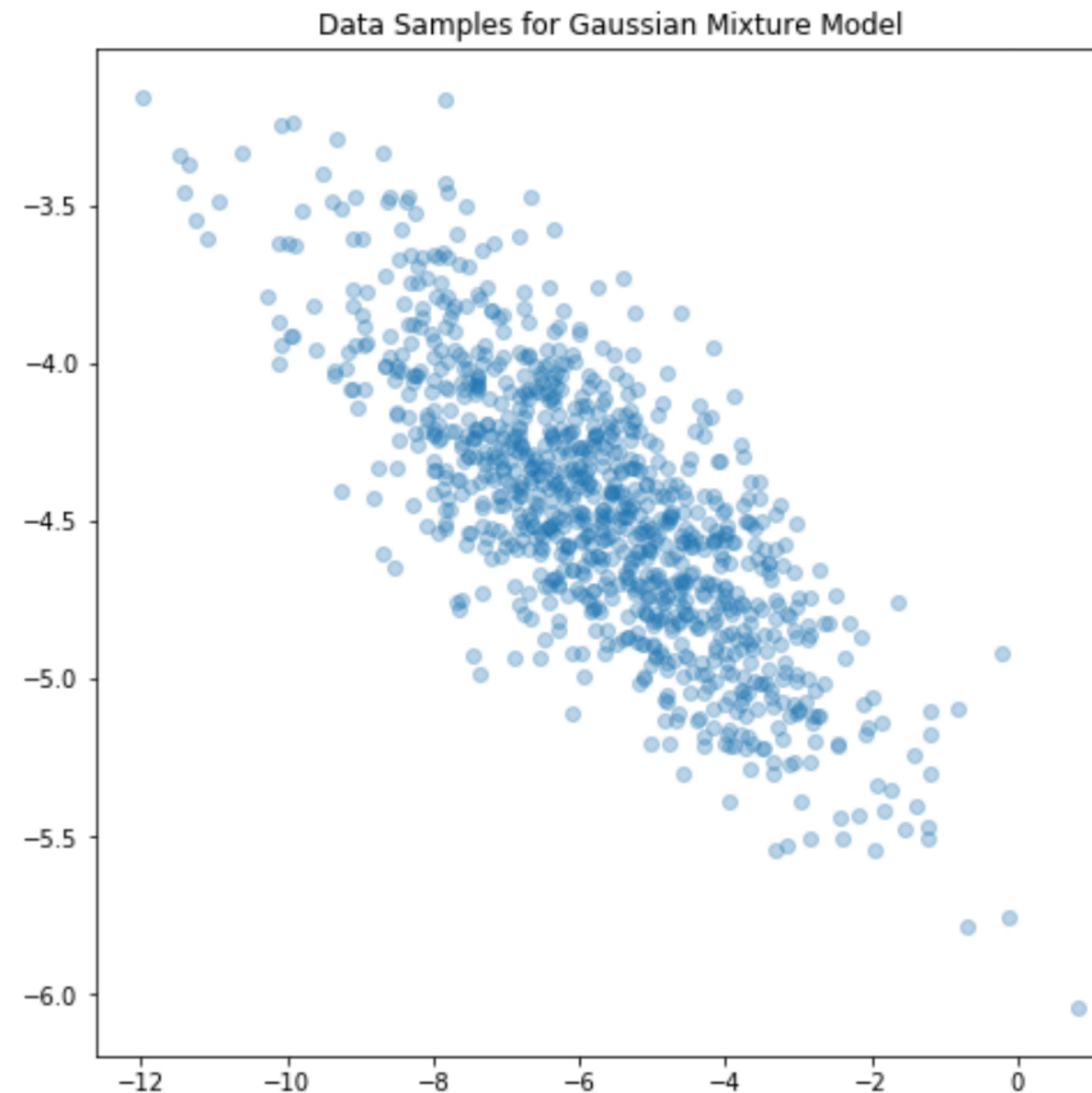
K-means

Figure 9.2 Plot of the cost function J given by (9.1) after each E step (blue points) and M step (red points) of the K -means algorithm for the example shown in Figure 9.1. The algorithm has converged after the third M step, and the final EM cycle produces no changes in either the assignments or the prototype vectors.



Gaussian Mixture Models

Gaussian Models



Gaussian Models

- Single multivariate Gaussian distribution을 이용해 dataset의 distribution을 모델링하고자 합니다.

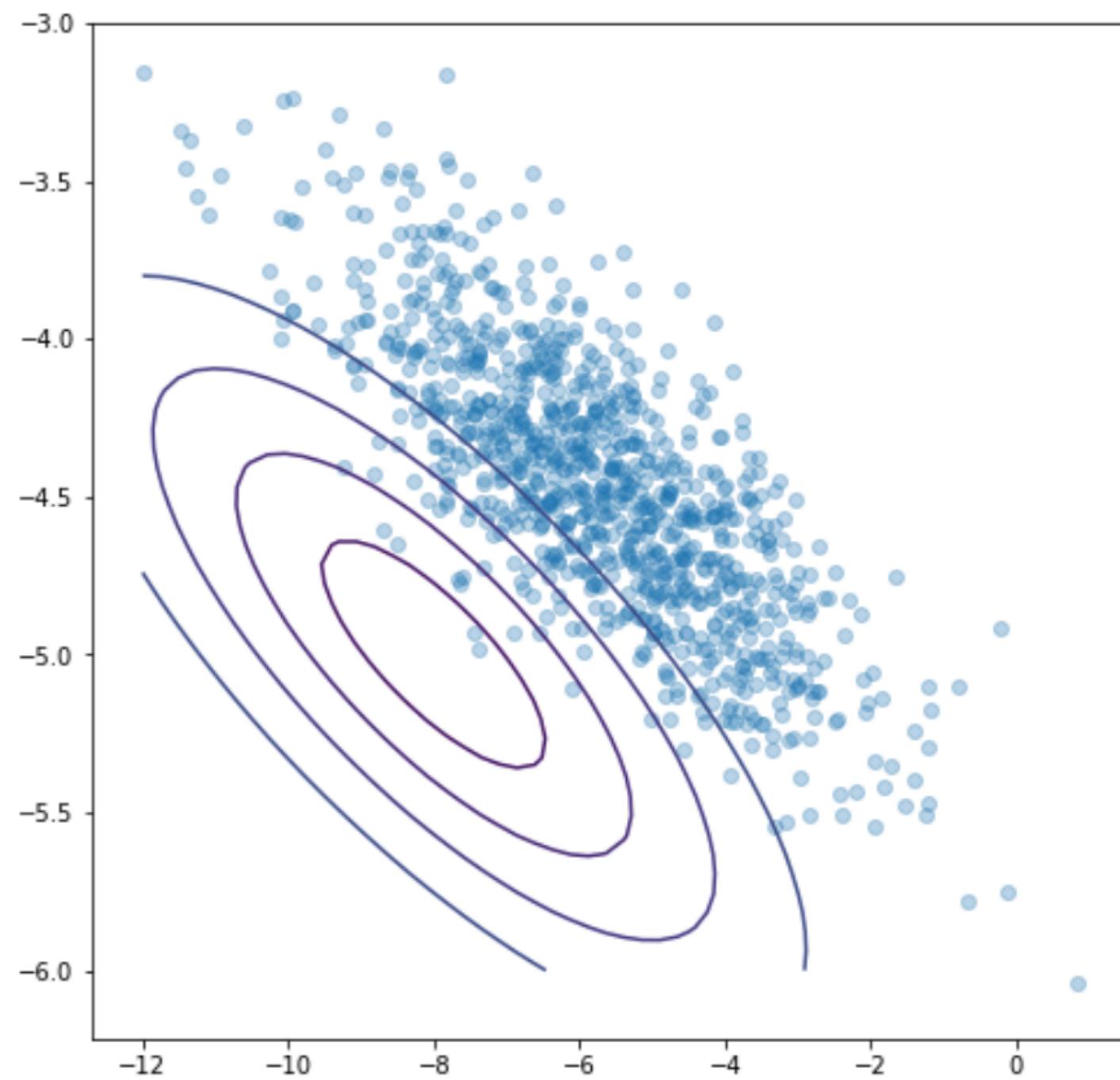
$$p_{\theta}(x) = N(x | \mu, \Sigma)$$

$$= \frac{1}{\sqrt{(2\pi)^D |\Sigma|}} \exp \left[-\frac{1}{2} (x - \mu) \Sigma^{-1} (x - \mu)^T \right]$$

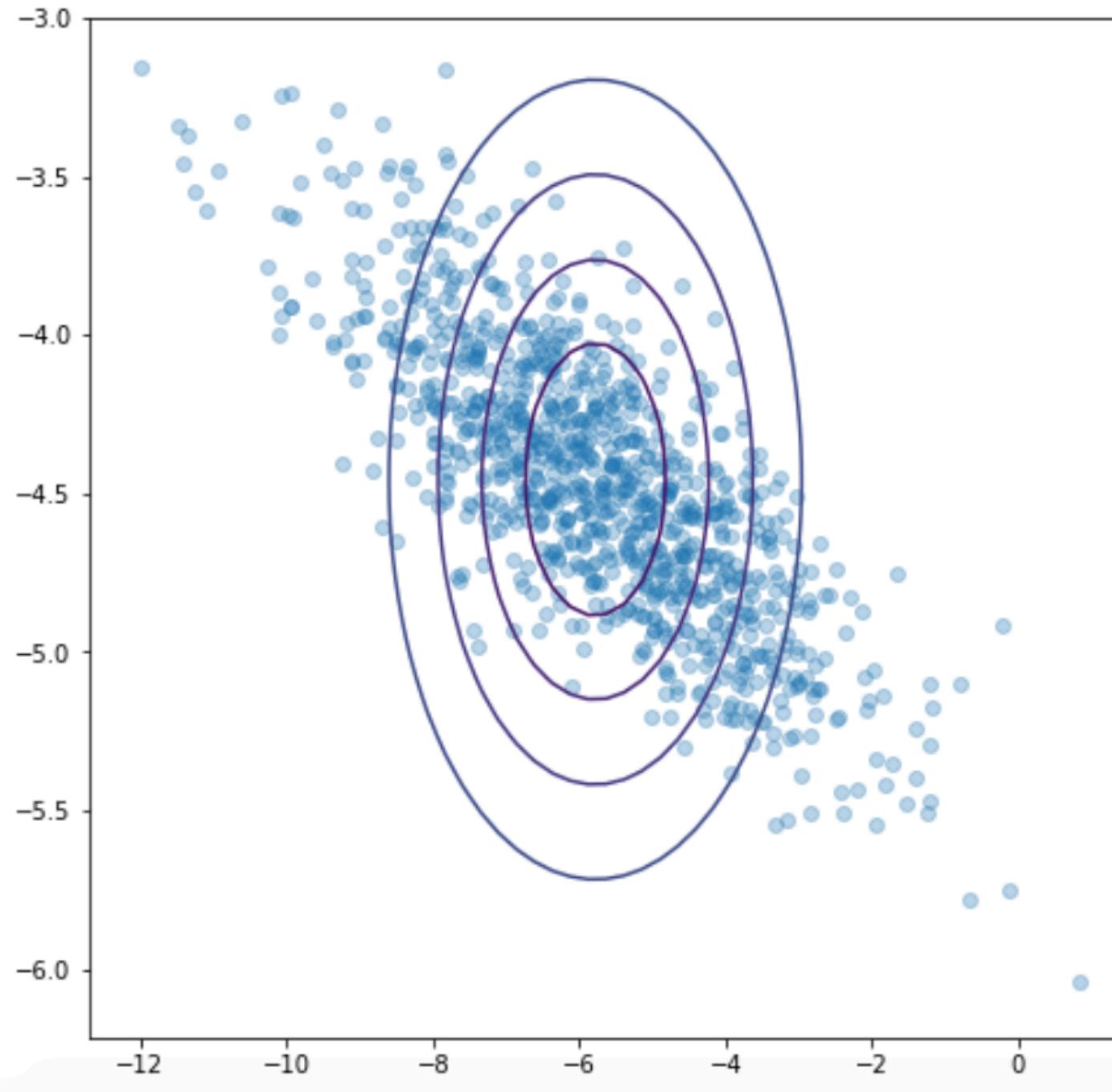
- 이를 위해서 maximum likelihood 방식을 이용합니다. 즉, dataset의 likelihood를 최대화하는 parameters $\theta = \{\mu, \Sigma\}$ 를 구합니다.

$$\arg \max_{\theta} p_{\theta}(X), \text{ where } X = \{x_1, x_2, \dots, x_N\}$$

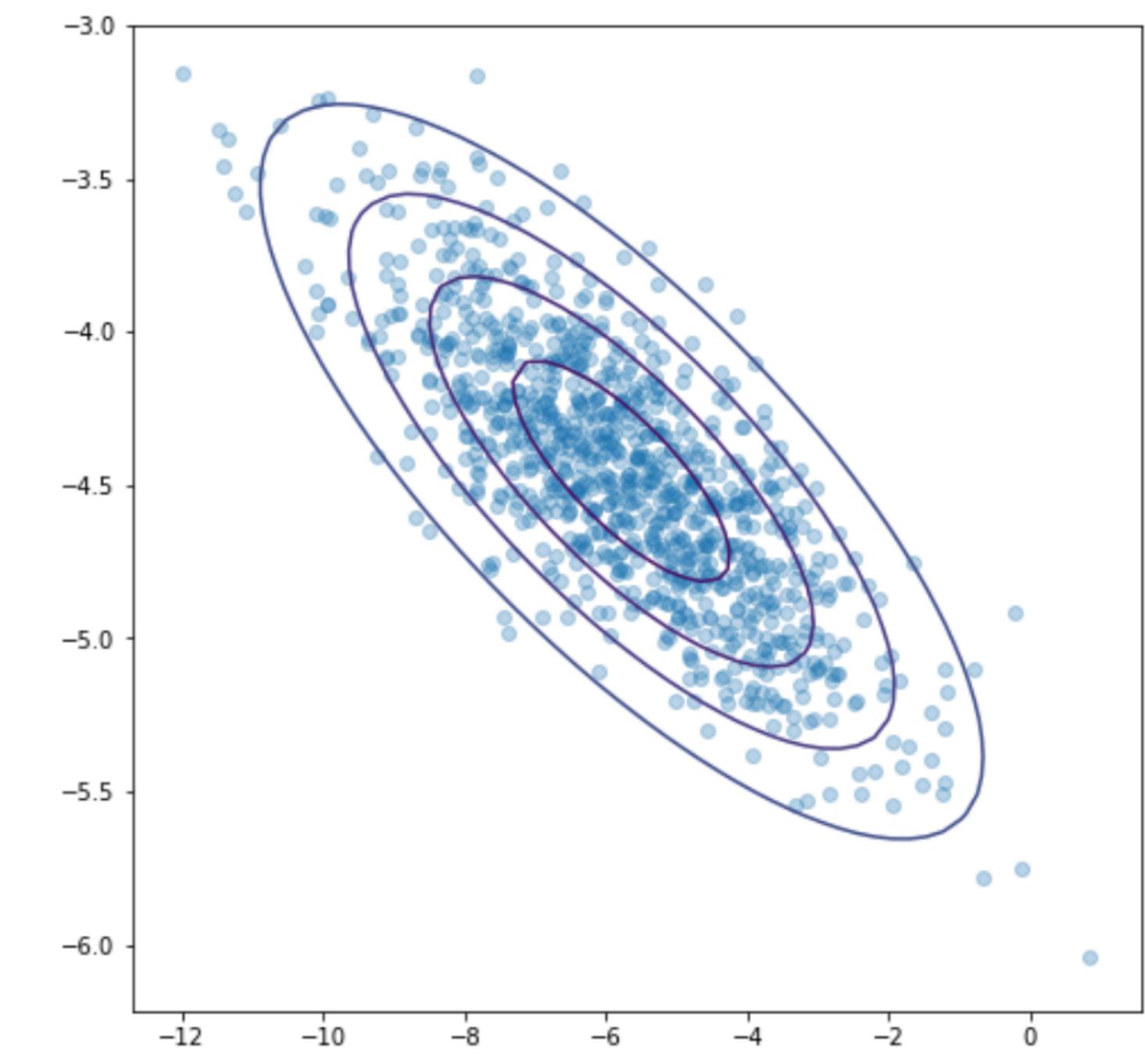
Gaussian Models



낮은 likelihood를 갖는 μ, Σ 가 맞춰진 상황



likelihood가 최대가 되도록 μ 가 맞춰졌으나 Σ 가 적합하지 않은 상황



likelihood가 최대가 되도록 μ 와 Σ 가 맞춰진 상황

Gaussian Models

- Distribution $p_\theta(x)$ 에서 data samples X 전체의 likelihood는 다음과 같습니다. 이 때, 각 samples들은 I.I.D. (independent and identically distributed)로 가정합니다.

$$p(X) = \prod_{n=1}^N p(x_n), X = \{x_1, x_2, \dots, x_N\}$$

- 위 식에 log를 씌워 Dataset 전체의 log-likelihood를 구하면 다음과 같습니다.

$$\log p_\theta(X) = \log \prod_{n=1}^N p_\theta(x_n) = \sum_{n=1}^N \log p_\theta(x_n)$$

- Single multivariate Gaussian distribution에서 data point 하나의 log-likelihood는 다음과 같습니다.

$$\log p_\theta(x_n) = -\frac{D}{2} \log 2\pi - \log |\Sigma| - \frac{1}{2}(x_n - \mu)\Sigma^{-1}(x_n - \mu)^T$$

- 따라서 dataset 전체의 log-likelihood는 다음과 같이 계산됩니다.

$$\log p_\theta(X) = \sum_{n=1}^N -\frac{D}{2} \log 2\pi - \log |\Sigma| - \frac{1}{2}(x_n - \mu)\Sigma^{-1}(x_n - \mu)^T$$

Gaussian Models

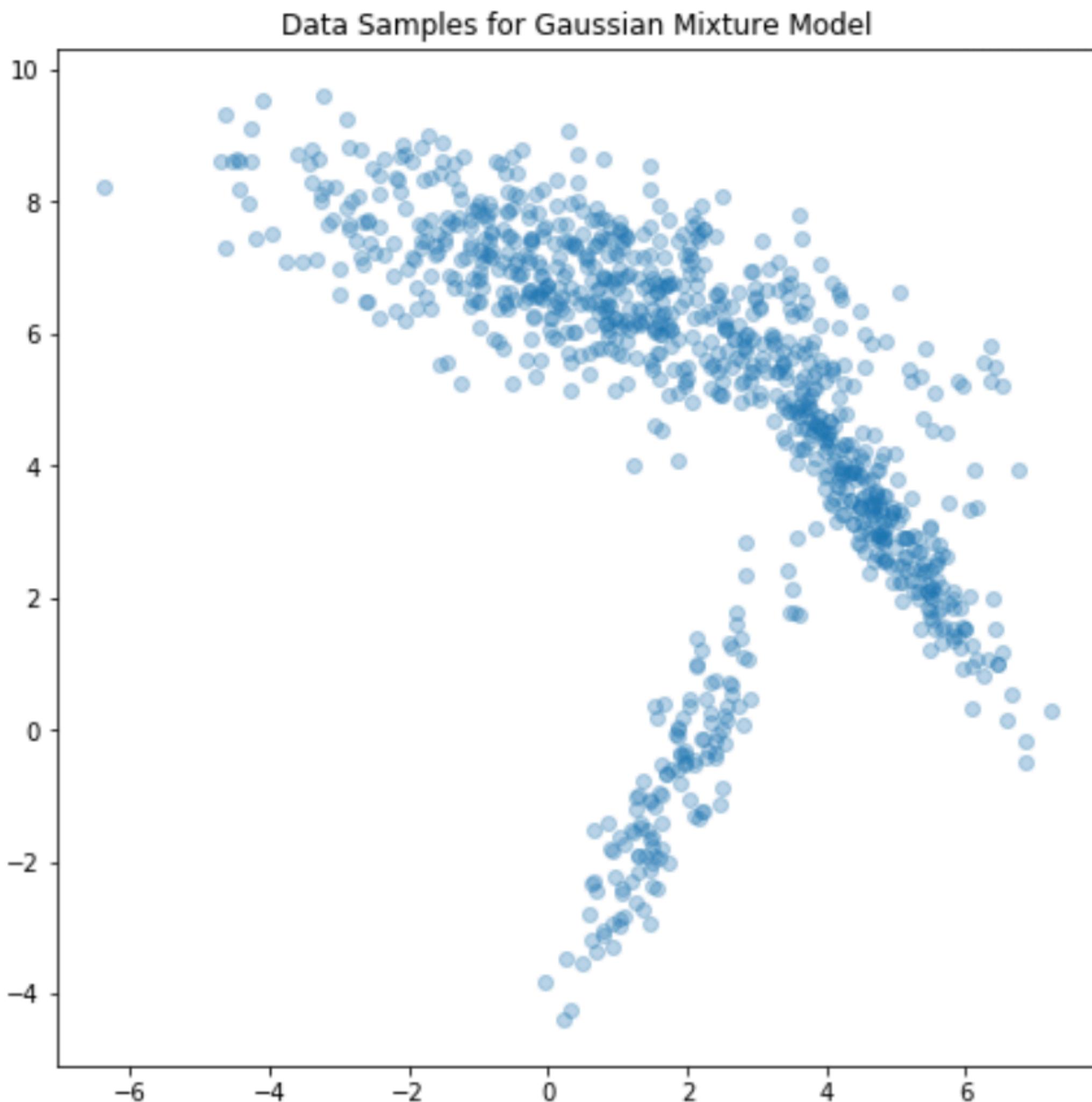
- $\log p_\theta(X)$ 가 최대가 되도록하는 parameters μ, Σ 를 다음과 같이 closed form solution을 구할 수 있습니다.

-

$$\mu_{ML} = \frac{1}{N} \sum_{n=1}^N x_n$$

$$\Sigma_{ML} = \frac{1}{N} \sum_{n=1}^N (x_n - \mu)(x_n - \mu)^T$$

Gaussian Mixture Models



Gaussian Mixture Models

- 여러 Multivariate Gaussian distributions를 이용해 dataset의 distribution을 모델링하고자 합니다.



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

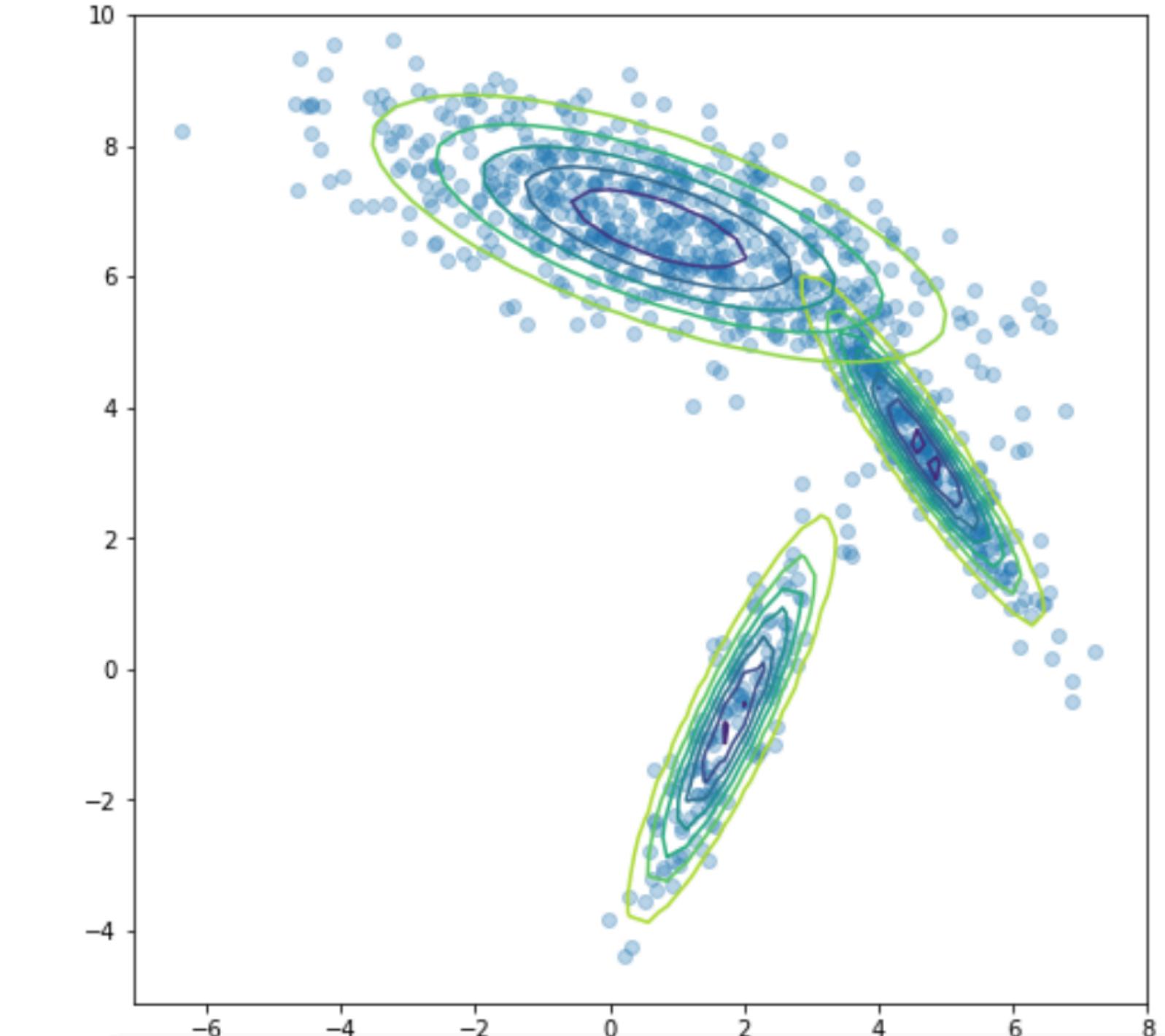
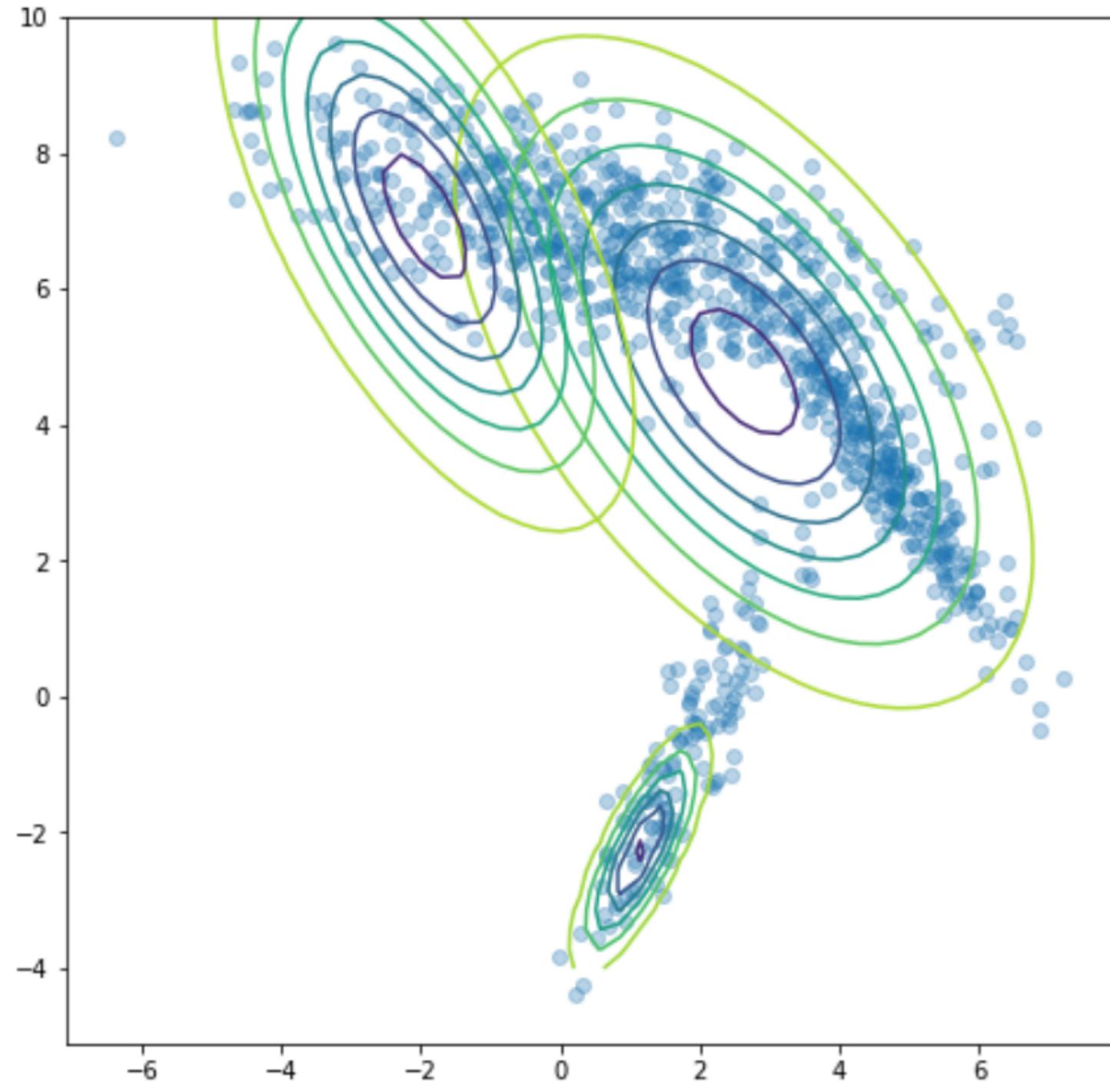
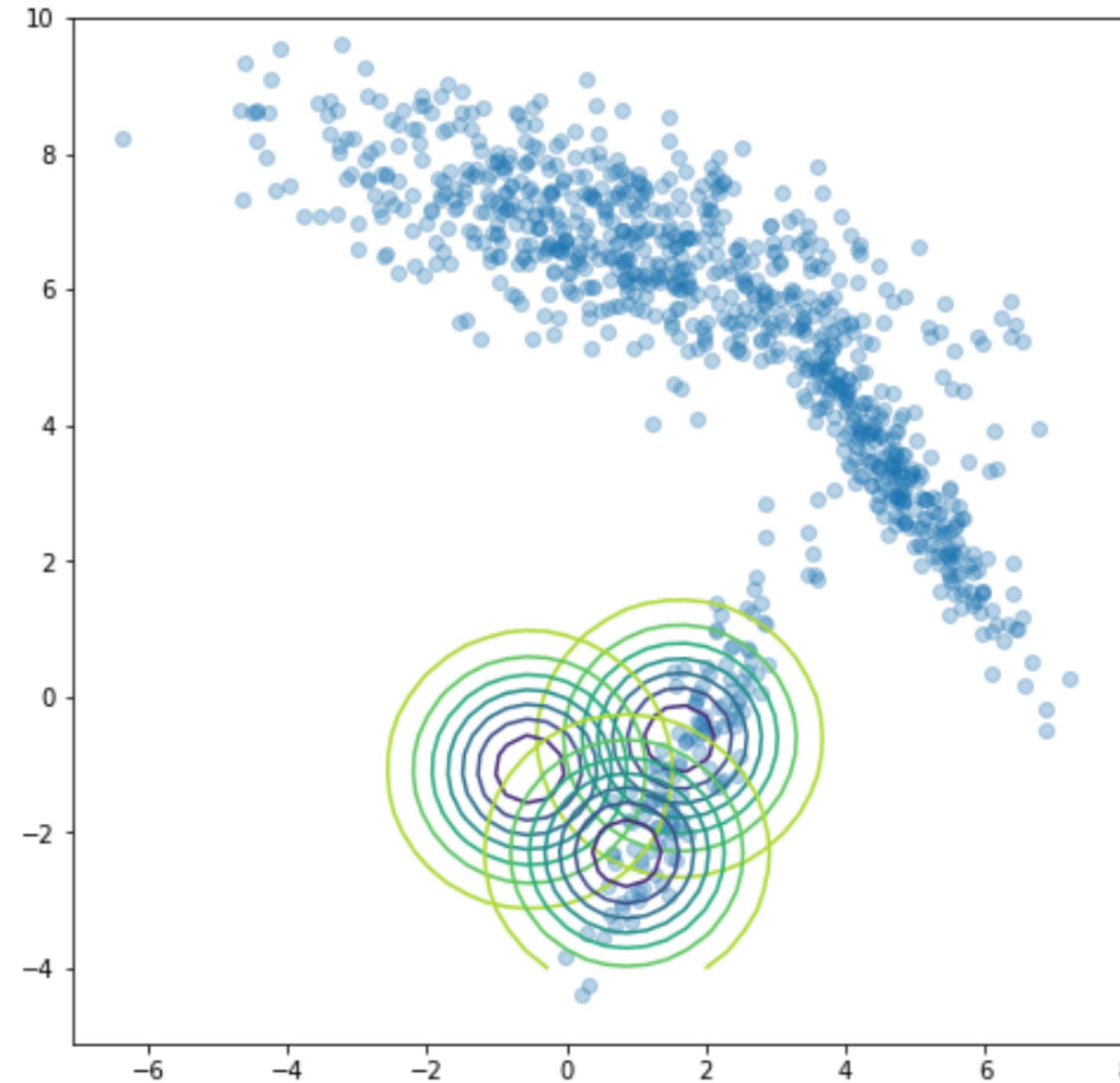
- 위 식에서 K 는 Gaussian distribution의 mixture 갯수를 뜻합니다. 각 Gaussian component 별로 parameters인 mixing coefficient π_k 와 mean μ_k , covariance matrix Σ_k 가 존재합니다.
- Maximum likelihood 방식을 이용하여 dataset에 맞는 distribution의 parameters를 찾습니다. 즉, dataset의 likelihood를 최대화하는 parameters $\theta = \{\pi_k, \mu_k, \Sigma_k\}$ 를 구합니다.



$$\arg \max_{\theta} p_{\theta}(X), \text{ where } X = \{x_1, x_2, \dots, x_N\}$$

Gaussian Mixture Models

- dataset의 likelihood가 커지도록 parameters가 업데이트 되는 모습



Gaussian Mixture Models

- Data point 하나는 다음과 같은 두 순서 의해서 생성되었다고 볼 수 있습니다.
 1. K 개의 components(Gaussian distributions) 중 하나를 선택
 2. 선택한 component에서 data point 샘플링
- Mixture의 각 component들이 선택될 확률을 categorical distribution으로 나타냅니다.
-
- 이 때, z 는 선택된 component를 나타내는 one-hot vector로 표현됩니다. 예를 들어 총 5개의 components가 있고 3번째 component가 선택되었다면 $z = (0,0,1,0,0)^T$ 를 갖게 됩니다. z_k 는 one-hot vector의 k 번째 element를 뜻합니다.
- Component가 정해졌을 때, 즉 z 가 주어졌을 때 data point의 likelihood는 Gaussian distribution으로 나타냅니다.

$$p_{\theta}(z) = \prod_{k=1}^K \pi_k^{z_k}$$

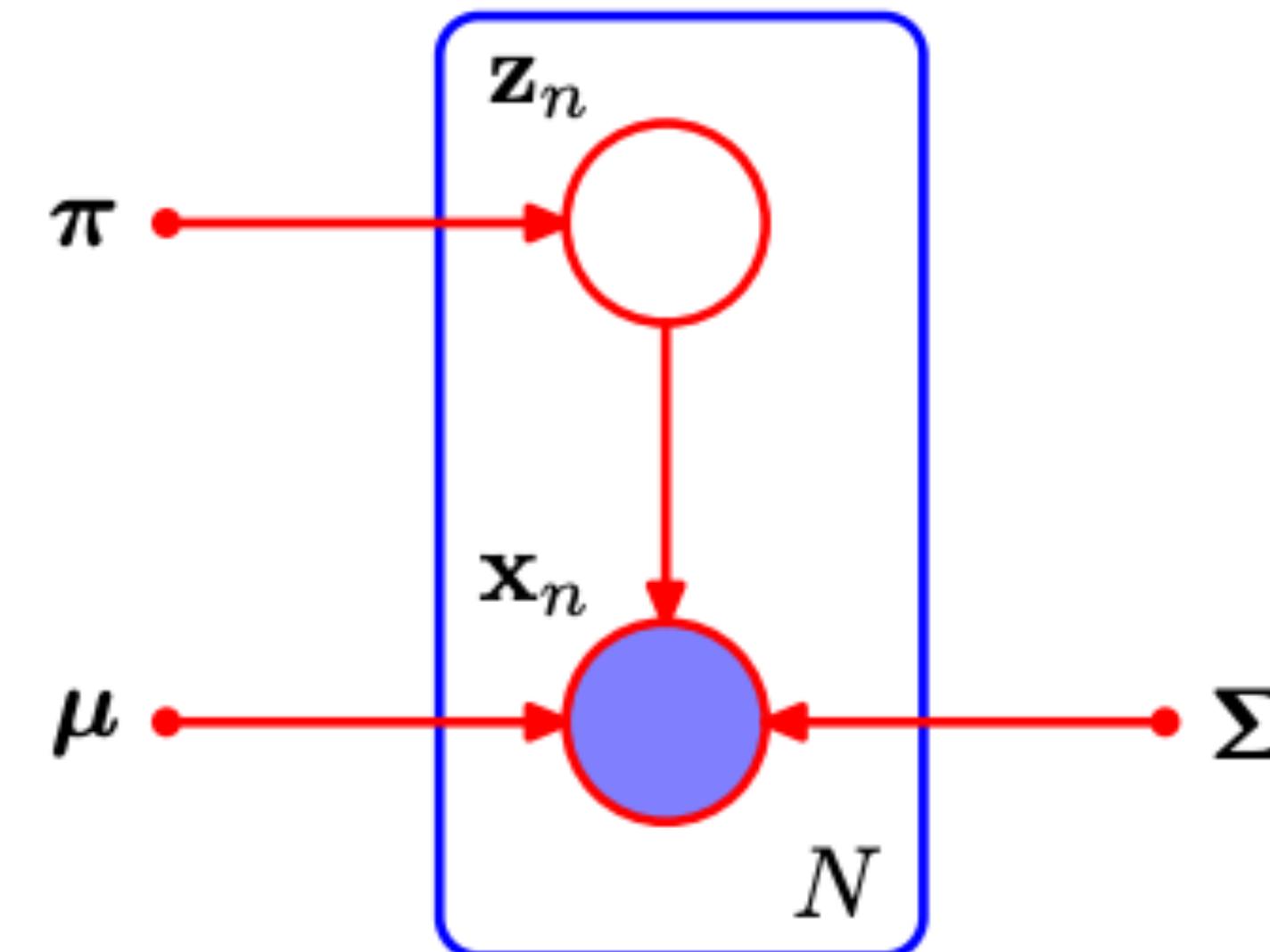
$$p_{\theta}(x | z) = \prod_{k=1}^K N(x | \mu_k, \Sigma_k)^{z_k}$$

Gaussian Mixture Models

- z 를 특정하지 않은 marginal likelihood는 다음과 같이 구할 수 있습니다.

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

- Gaussian mixture models을 graphical representation으로 나타낼 수 있습니다.



Gaussian Mixture Models

- Single Gaussian Model의 경우와 마찬가지로 I.I.D. dataset 전체의 likelihood는 다음과 같습니다.

$$p_{\theta}(X) = \prod_{n=1}^N p_{\theta}(x_n), X = \{x_1, x_2, \dots, x_N\}$$

- Gaussian mixture models 경우 log-likelihood는 다음과 같이 전개됩니다.

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

- Single Gaussian model과 다르게 log-likelihood를 최대화하는 parameters를 closed form으로 구할 수 없습니다.

Gaussian Mixture Models

- log-likelihood가 최대가 되는 mean μ_k 을 구해보기 위해 μ_k 로 미분하고 이를 0으로 두면 다음과 같습니다.

-

$$0 = \sum_{n=1}^N \frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j N(x_n | \mu_j, \Sigma_j)} \Sigma^{-1}(x_n - \mu_k)$$

- 만약 위 식에서 $\frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j N(x_n | \mu_j, \Sigma_j)}$ 부분을 어떤 값으로 고정한다면 μ_k 를 구할 수 있습니다.

$$0 = \sum_{n=1}^N \gamma(z_{nk}) \Sigma^{-1}(x_n - \mu_k)$$

Σ 를 양변에 곱하면

$$0 = \sum_{n=1}^N \gamma(z_{nk}) x_n - \sum_{n=1}^N \gamma(z_{nk}) \mu_k$$

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) x_n, \text{ where } N_k = \sum_{n=1}^N \gamma(z_{nk})$$

Gaussian Mixture Models

- 앞에서 고정한 값은 사실 어떤 data point x_n 이 주어졌을 때 k 번째 component로부터 생성되었을 확률을 나타내는 posterior로 해석할 수 있습니다. GMM에서 이를 responsibility라고 말하기도 합니다.

$$0 = \sum_{n=1}^N \frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j N(x_n | \mu_j, \Sigma_j)} \Sigma^{-1}(x_n - \mu_k)$$

↓

Prior $p(z_k = 1)$ Likelihood $p(x_n | z_k = 1)$

Marginal Likelihood(Evidence)

$$\sum_{z_j} p(z_j = 1) p(x_n | z_j = 1) = p(x_n)$$

$$\text{Prior} \times \text{Likelihood} / \text{Evidence} = \text{Posterior}$$
$$p(z_{nk} = 1 | x_n)$$

Gaussian Mixture Models

- posterior를 고정하면 mean μ_k 을 비롯해 mixing coefficient π_k 와 covariance matrix Σ_k 도 closed form으로 구할 수 있습니다.

$$\bullet \quad \pi_k = \frac{N_k}{N},$$

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) x_n,$$

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T$$

$$\text{where } \gamma(z_{nk}) = \sum_{n=1}^N \frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j N(x_n | \mu_j, \Sigma_j)} \text{ and } N_k = \sum_{n=1}^N \gamma(z_{nk})$$

Gaussian Mixture Models

- Parameters $\theta = \{\pi_k, \mu_k, \Sigma_k\}$ 를 구하고 나면 그에 따라 posterior인 responsibility $\gamma(z_{nk})$ 값도 변하게 됩니다. 따라서 EM 알고리즘이라 불리는 iteration을 통해 최적화를 진행하게 됩니다.

- Expectation-Maximization Algorithm for GMM

1. parameters $\theta = \{\pi_k, \mu_k, \Sigma_k\}$ 를 임의의 값으로 초기화 한다.

2. E-step : responsibility를 구한다.

$$\gamma(z_{nk}) = \frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j N(x_n | \mu_j, \Sigma_j)}$$

3. M-step : responsibility를 고정하고 parameters를 다시 구한다.

$$\pi_k^{new} = \frac{N_k}{N}, \mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) x_n, \Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (x_n - \mu_k^{new})(x_n - \mu_k^{new})^T$$

$$where N_k = \sum_{n=1}^N \gamma(z_{nk})$$

4. 정해진 step을 돌거나, 정해놓은 수렴 기준에 도달할 때 까지 E-step과 M-step을 반복한다.

Gaussian Mixture Models

- 현재 가지고 있는 dataset은 $X = \{x_1, x_2, \dots, x_N\}$ 입니다. 이를 incomplete dataset이라고 합니다.
- 모든 X 에 대응하는 모든 latent variables $Z = \{z_1, z_2, \dots, z_N\}$ 도 가지고 있다면 이를 complete dataset이라고 할 수 있습니다.

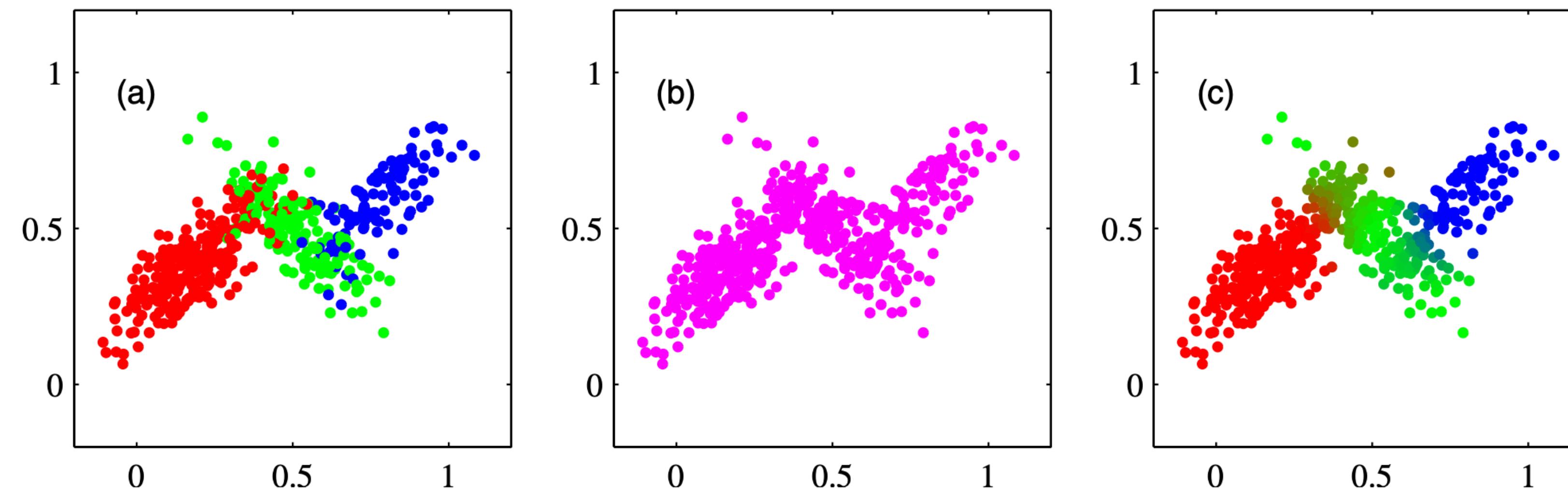


Figure 9.5 Example of 500 points drawn from the mixture of 3 Gaussians shown in Figure 2.23. (a) Samples from the joint distribution $p(z)p(x|z)$ in which the three states of z , corresponding to the three components of the mixture, are depicted in red, green, and blue, and (b) the corresponding samples from the marginal distribution $p(x)$, which is obtained by simply ignoring the values of z and just plotting the x values. The data set in (a) is said to be *complete*, whereas that in (b) is *incomplete*. (c) The same samples in which the colours represent the value of the responsibilities $\gamma(z_{nk})$ associated with data point x_n , obtained by plotting the corresponding point using proportions of red, blue, and green ink given by $\gamma(z_{nk})$ for $k = 1, 2, 3$, respectively

Gaussian Mixture Models

- Dataset으로 $X = \{x_1, x_2, \dots, x_N\}$ 와 $Z = \{z_1, z_2, \dots, z_N\}$ 가 주어진 경우 complete data log-likelihood를 다음과 같이 쓸 수 있습니다.

$$\log p_\theta(X, Z) = \log p_\theta(Z) + \log p_\theta(X | Z) = \sum_{n=1}^N [\log p_\theta(z_n) + \log p_\theta(x_n | z_n)]$$

- $p_\theta(z)$ 는 categorical distribution, $p_\theta(x | z)$ 는 Gaussian distribution이므로 complete data-log-likelihood를 최대화하는 것은 앞에서 보인 것처럼 쉽게 전개할 수 있습니다.

- $\log p_\theta(Z)$ 를 최대화하는 parameter π_k 를 구하면 다음과 같습니다.

$$\pi_k = \frac{N_k}{N}, \text{ where } N_k = \sum_{n=1}^N z_{nk}$$

- Z 가 주어졌을 때 $\log p_\theta(X | Z)$ 를 최대화하는 parameters μ_k, Σ_k 를 구하면 다음과 같습니다.

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^N z_{nk} x_n,$$

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^N z_{nk} (x_n - \mu_k)(x_n - \mu_k)^T$$

Gaussian Mixture Models

Posterior(responsibility)가 주어졌을 때
EM 알고리즘의 M-step

$$\pi_k = \frac{N_k}{N},$$

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) x_n,$$

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T$$

$$\text{where } \gamma(z_{nk}) = \sum_{n=1}^N \frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j N(x_n | \mu_j, \Sigma_j)}$$

$$\text{and } N_k = \sum_{n=1}^N \gamma(z_{nk})$$

Complete dataset X, Z 가 주어졌을 때
complete data log-likelihood의 최대화

$$\pi_k = \frac{N_k}{N},$$

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^N z_{nk} x_n,$$

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^N z_{nk} (x_n - \mu_k)(x_n - \mu_k)^T$$

$$\text{where } N_k = \sum_{n=1}^N z_{nk}$$

Gaussian Mixture Models

- 앞서 EM 알고리즘을 통해 구한 방법은 posterior $p_{\theta}(Z|X)$ 로 가상의 데이터 z-variables를 만들고 complete data log-likelihood를 최대화한 것과 같다고 할 수 있습니다.
- 따라서 E-step과 M-step은 다음과 같이 일반할 수 있습니다.

E-step : dataset 전체의 posterior $p_{\theta}(Z|X)$ 를 구한다.

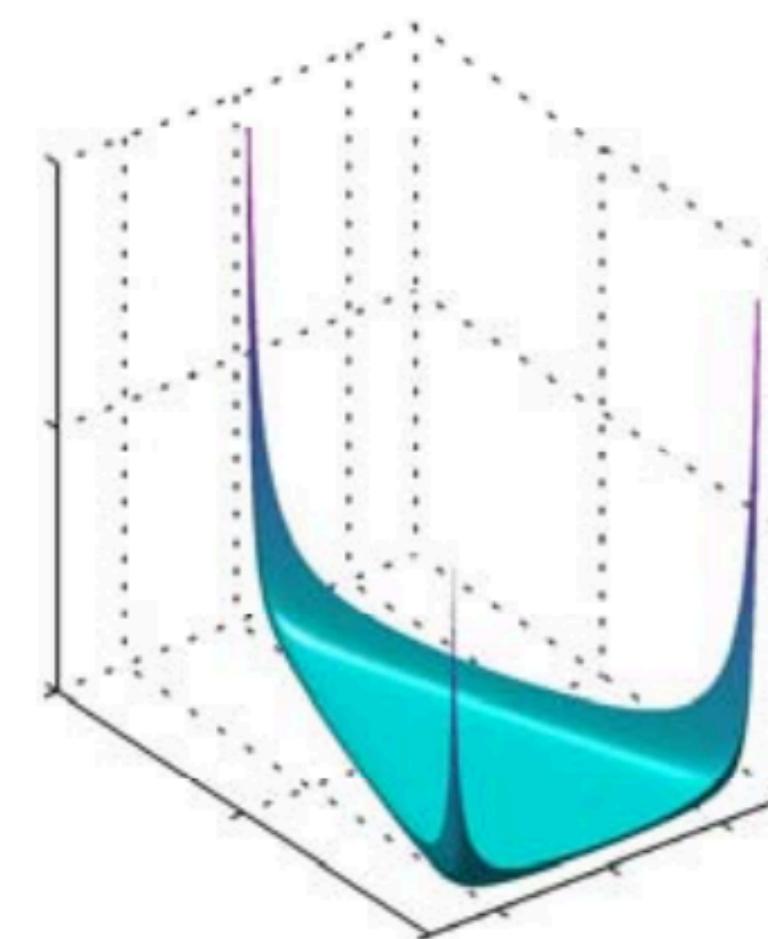
M-step : posterior에 의한 complete data log-likelihood의 기댓값을 최대화하는 parameters를 구한다.

$$\theta^{new} = \arg \max_{\theta} \mathbb{E}_{p_{\theta}(Z|X)} [\log p_{\theta^{old}}(X, Z)]$$

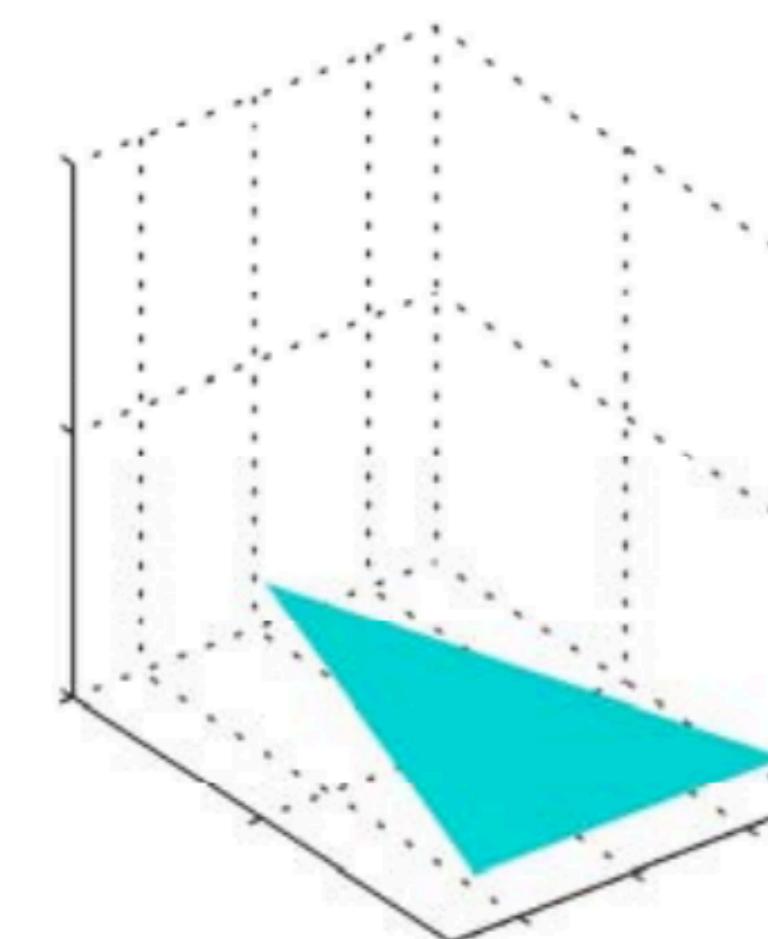
Variational Gaussian Mixture Models

Variational Gaussian Mixture Models

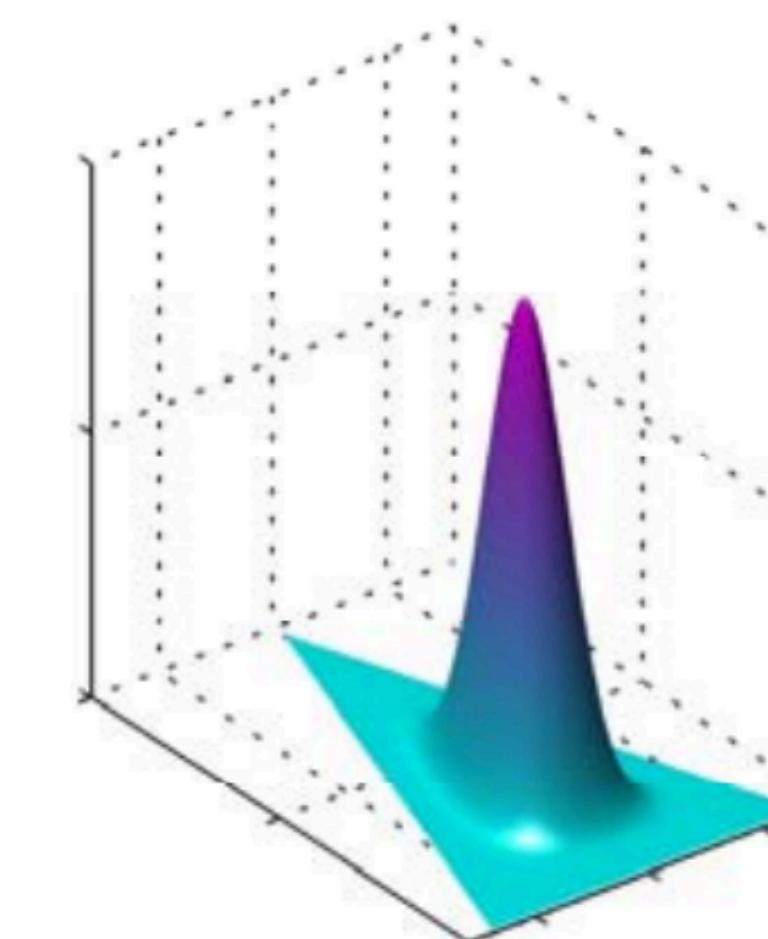
- Gaussian Mixture Models에서 parameters로 사용되었던 mixing coefficients π_k , mean μ_k , covariance matrix Σ_k 를 random variables로 여기고 probability distribution을 갖도록 모델링 합니다.
- Mixing coefficients π_k 에 대한 prior는 Dirichlet로 정하고 이의 concentration parameter $\alpha_1, \dots, \alpha_K$ 를 모두 α_0 로 일치시킵니다. 또한 mean μ_k 과 covariance matrix Σ_k 에 대한 prior는 Gaussian-Wishart로 정합니다.



$$\alpha_0 = 0.1$$

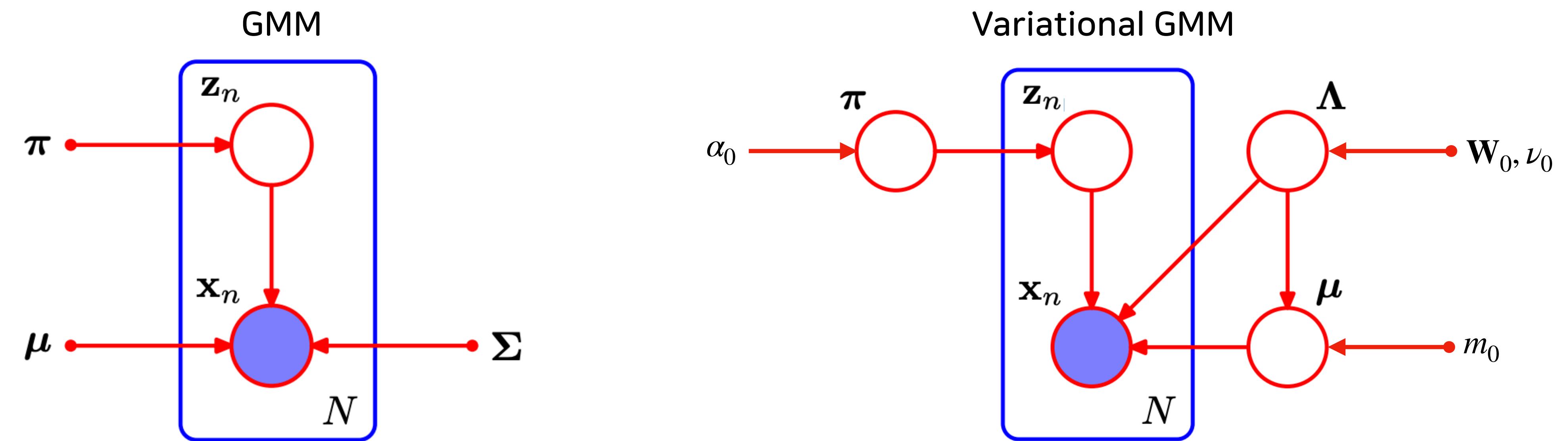


$$\alpha_0 = 1$$



$$\alpha_0 = 10$$

Variational Gaussian Mixture Models

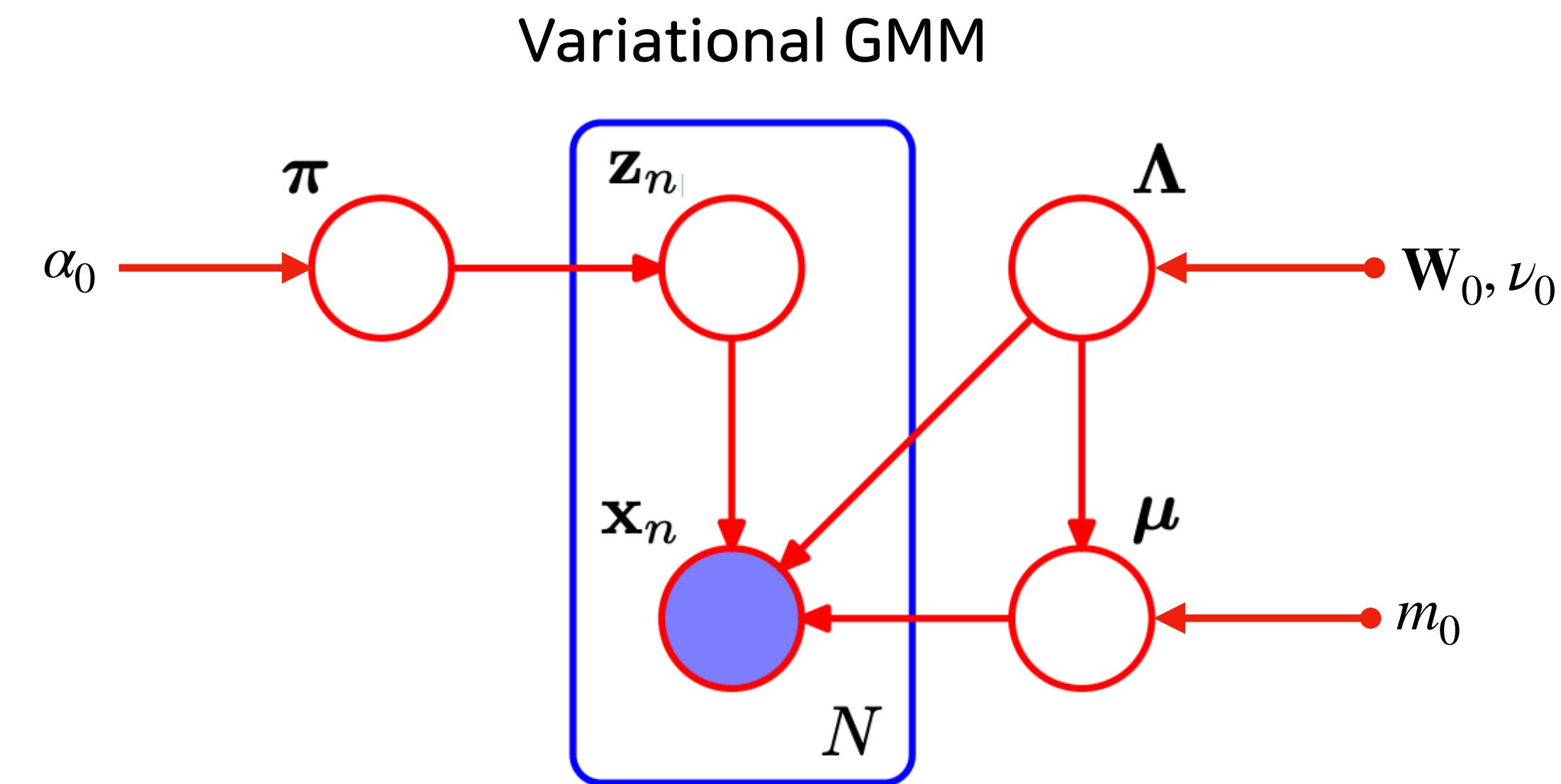
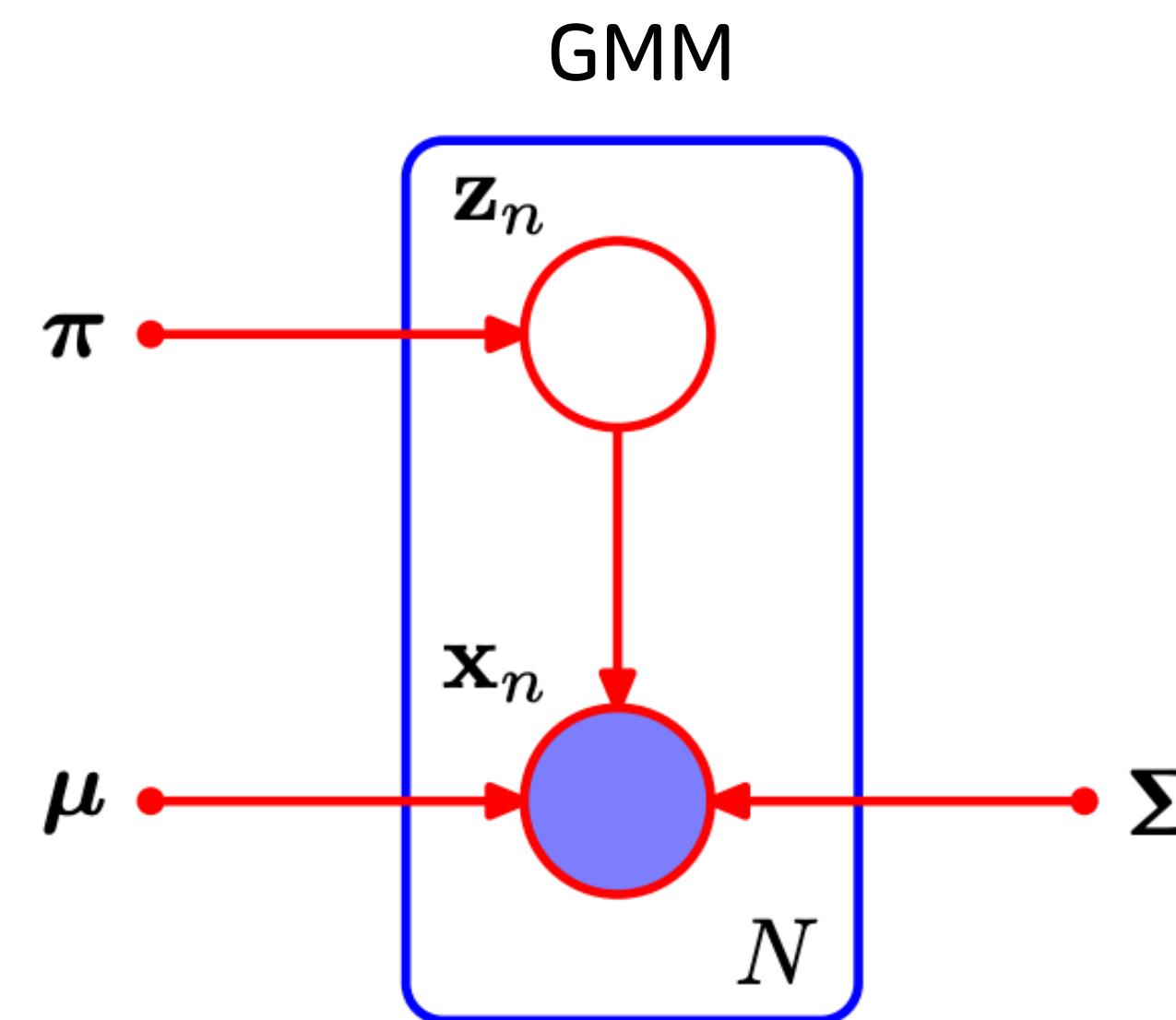


Mixing Coefficients $p_\theta(z) = \prod_{k=1}^K \pi_k^{z_k}$

$$p_\theta(z | \pi) = \prod_{k=1}^K \pi_k^{z_k}$$

Dirichlet Prior $p_\theta(\pi) = Dir(\pi | \alpha_0) = C(\alpha_0) \prod_{k=1}^K \pi_k^{\alpha_k - 1}$

Variational Gaussian Mixture Models



Conditional distribution

$$p_\theta(z) = \prod_{k=1}^K \pi_k^{z_k}$$

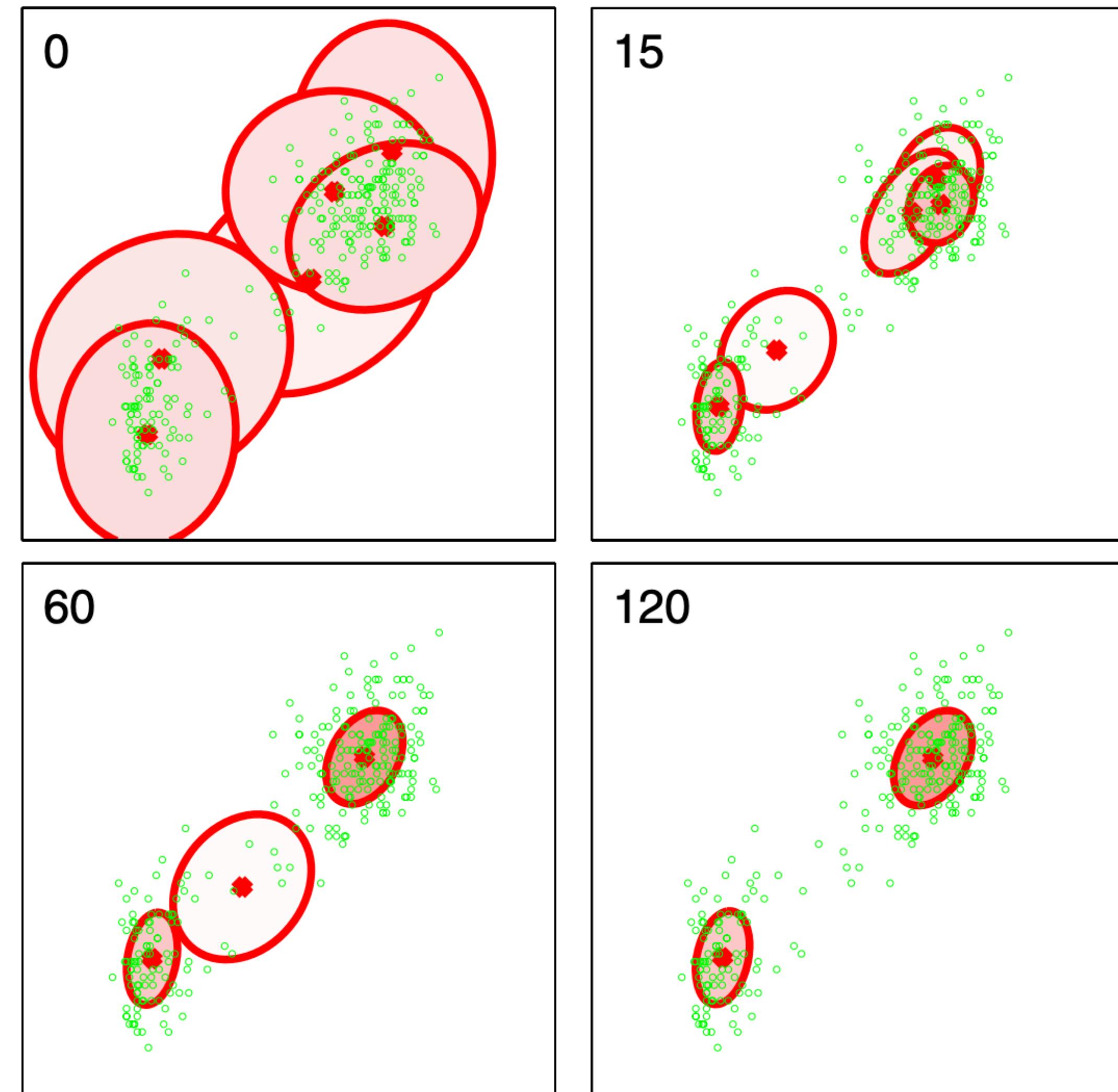
$$p_\theta(x | z, \mu, \Lambda) = \prod_{k=1}^K N(x | \mu_k, \Lambda_k^{-1})^{z_k}$$

Gaussian-Wishart Prior $p(\mu, \Lambda) = p(\mu | \Lambda)p(\Lambda)$

$$= \prod_{k=1}^K \mathcal{N}(\mu_k | \mathbf{m}_0, (\beta_0 \Lambda_k)^{-1}) \mathcal{W}(\Lambda_k | \mathbf{W}_0, \nu_0)$$

Variational Gaussian Mixture Models

- Mixing coefficients π_k 에 대한 prior를 Dirichlet로 두고 concentration parameter α_0 을 낮은 값($1e-3$)으로 설정하면, EM 알고리즘이 진행함에 따라 component 갯수가 적게 변하는 것을 볼 수 있습니다.

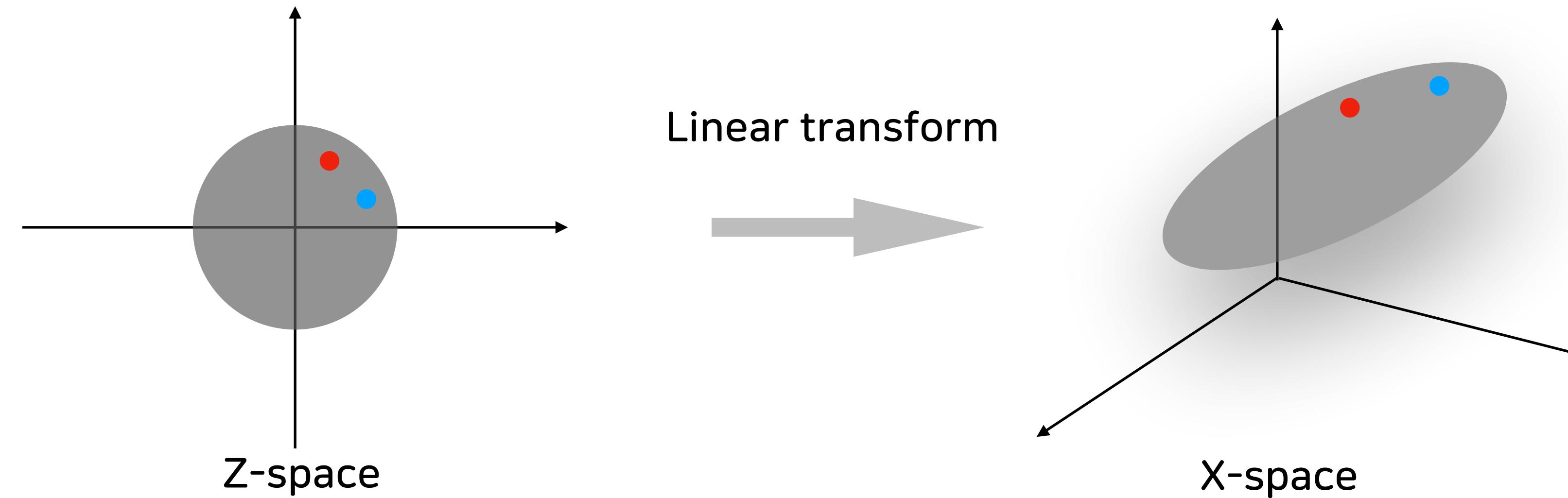


Probabilistic PCA

Probabilistic PCA

- PPCA(Probabilistic PCA)는 dimension reduction에 사용되는 PCA의 (semi) Bayesian 버전입니다. parameters를 random variables로 두지 않았으므로 fully Bayesian이라고 볼 수 없습니다.
- GMM에서 discrete한 z-variable을 갖는데 반해, PPCA에서는 continuous한 z-variable을 갖습니다.
- GMM에서 data point 하나를 얻기 위해 다음과 같은 순서에 의해 샘플링 하였습니다.
 1. K 개의 components(Gaussian distributions) 중 하나를 선택
 2. 선택한 component에서 data point 샘플링
- PPCA에서는 다음과 같은 순서에 의해 샘플링이 이루어집니다.
 1. Z-space의 Gaussian distribution에서 샘플 하나를 선택
 2. 샘플을 linear transform을 통해 X-space로 맵핑
 3. 맵핑된 값을 mean으로 하는 또 다른 Gaussian distribution에서 data point 샘플링

Probabilistic PCA



Prior $p_\theta(z) = N(z | 0, I)$

Conditional $p_\theta(x | z) = N(x | Wz + \mu, \sigma^2 I)$

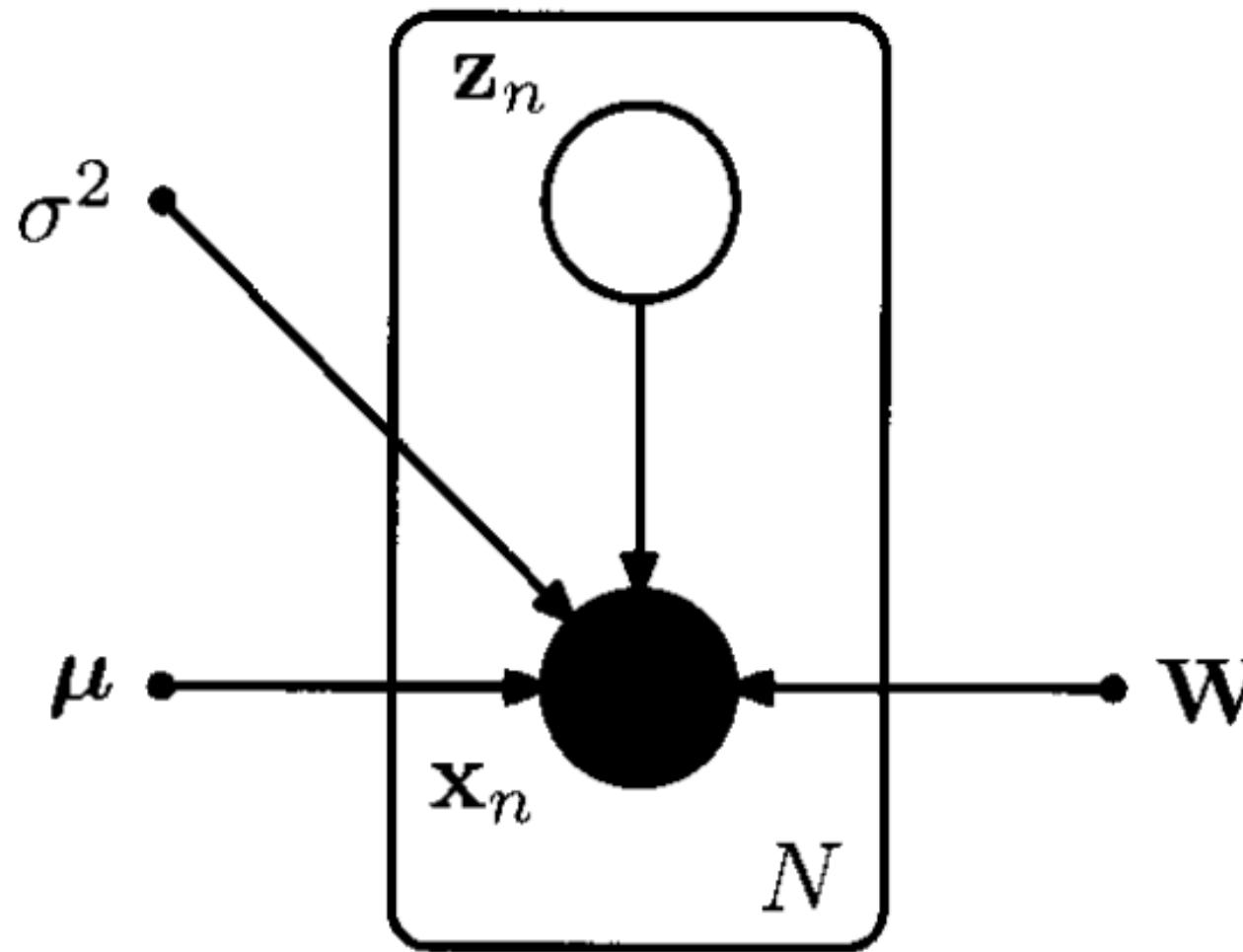
(Linear-Gaussian Model)

References :

Michael E. Tipping and Christopher M. Bishop. Probabilistic Principal Component Analysis. 1999.
Christopher M. Bishop. Pattern Recognition and Machine Learning. p.570-580

Probabilistic PCA

Graphical Representation



Prior $p_{\theta}(z) = N(z | 0, I)$

Conditional $p_{\theta}(x | z) = N(x | Wz + \mu, \sigma^2 I)$

(Linear-Gaussian Model)

Probabilistic PCA

- Maximum likelihood를 통해 parameters를 구하기 위해 marginal likelihood를 구합니다.

$$\begin{aligned} p_{\theta}(x) &= \int p_{\theta}(x, z) dz = \int p_{\theta}(z)p_{\theta}(x|z) dz = \int N(z|0, I)N(x|Wz + \mu, \sigma^2 I) dz \\ &= N(x|\mu, C), \text{ where } C = WW^T + \sigma^2 I \end{aligned}$$

- 위 식을 통해 dataset X 의 marginal log-likelihood를 구합니다.

$$\log p_{\theta}(X) = \sum_{n=1}^N \log p_{\theta}(x_n)$$

$$= -\frac{ND}{2} \log(2\pi) - \frac{N}{2} \log |C| - \frac{1}{2} \sum_{n=1}^N (x_n - \mu)^T C^{-1} (x_n - \mu)$$

$$p(x) = N(x|\mu, \Lambda^{-1})$$

$$p(y|x) = N(y|Ax + b, L^{-1})$$

$$p(y) = N(y|A\mu + b, L^{-1} + A\Lambda^{-1}A^T)$$

PRML p.93

Probabilistic PCA Maximum Likelihood

- GMM에서와 다르게 marginal likelihood가 Gaussian이 되고, 이를 최대화하는 parameters를 다음과 같이 closed form으로 구할 수 있습니다.

- $W = U_M(L_M - \sigma^2 I)^{1/2}R$ $U_M L_M U_M^T = S$ (*data covariance matrix*)
 $U_M = [v_1 \ v_2 \ \cdots \ v_M]$, where v_1, v_2, \dots, v_M = eigenvectors given by the data covariance matrix

$$L_M = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_M \end{bmatrix}, \text{where } \lambda_1, \lambda_2, \dots, \lambda_M = \text{the corresponding eigenvalues}$$

R_M = an arbitrary $M \times M$ orthogonal matrix

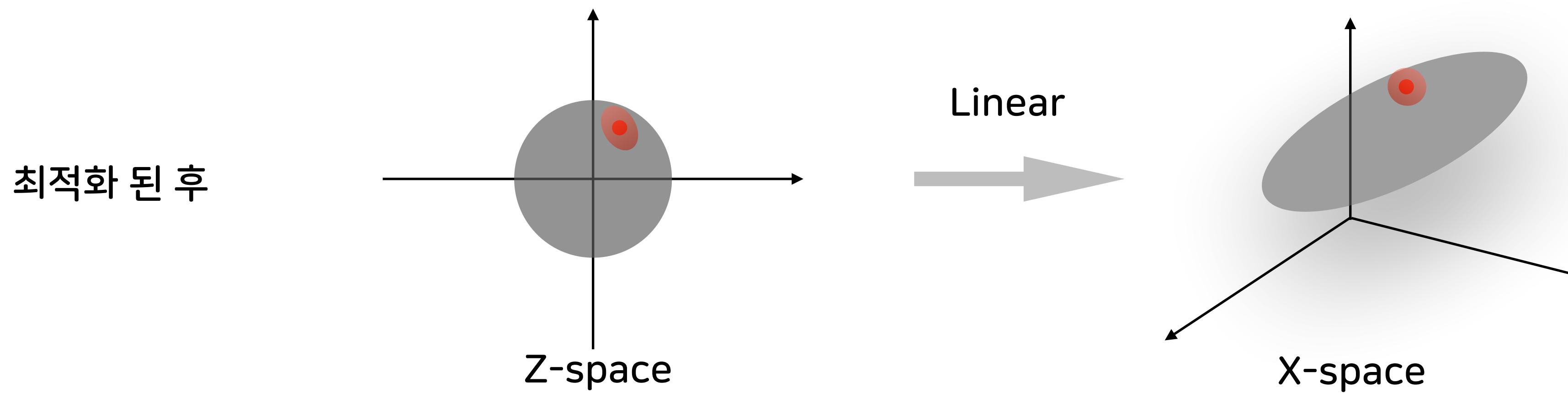
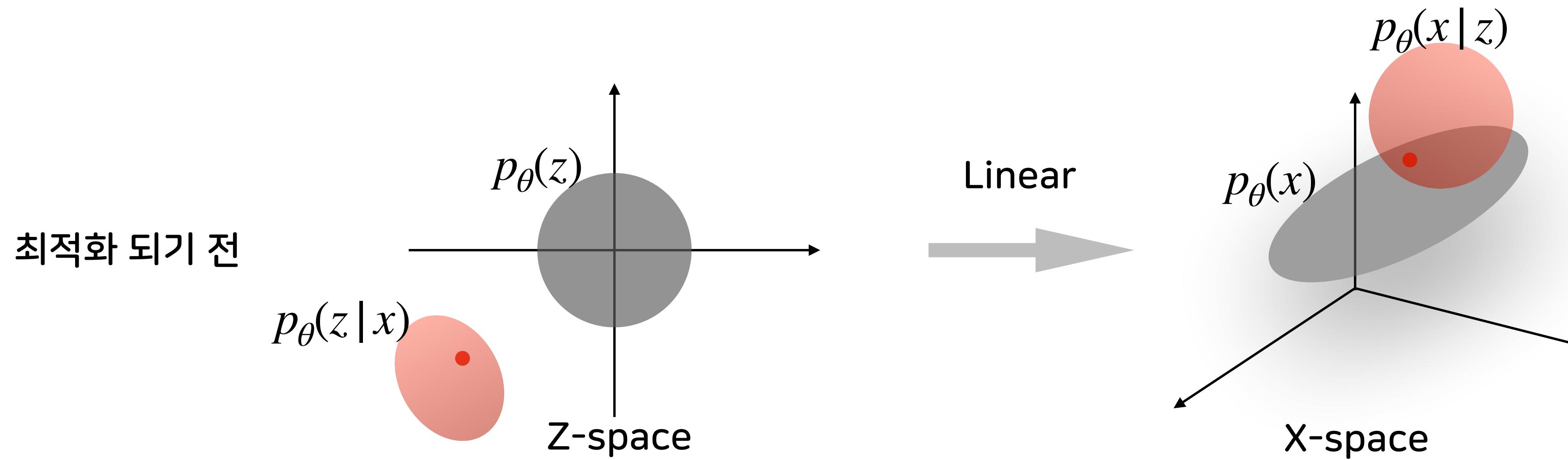
- $\mu = \frac{1}{N} \sum_{n=1}^N x_n$

- $\sigma^2 = \frac{1}{D-M} \sum_{i=M+1}^D \lambda_i$

Probabilistic PCA Expectation-Maximization

- 또는 GMM에서 했듯이 EM알고리즘으로 iteration을 통해 구할 수도 있습니다.
- PPCA에서 sampling은 두 번 이루어집니다.
 1. Prior $p_\theta(z)$ 에서 sampling
 2. Linear transform후 얻어진 conditional distribution $p_\theta(x | z)$ 에서 sampling
- 위 sampling을 통해 가지고 있는 dataset이 잘 나올 수 있도록 하기 위해서는 다음과 같은 두가지 조건을 만족해야 합니다.
 1. posterior $p_\theta(z | x)$ 에서 뽑은 sample이 prior $p_\theta(z)$ 에서 높은 likelihood를 가져야 한다.
 2. posterior $p_\theta(z | x)$ 에서 뽑은 sample로 linear transform을 시킨 후 얻은 conditional distribution $p_\theta(x | z)$ 에서 가지고 있는 dataset이 높은 likelihood를 가져야 한다.

Probabilistic PCA Expectation-Maximization



Probabilistic PCA Expectation-Maximization

- 위 두가지 조건을 만족하도록 하는 parameters를 구하기 위해 수식을 전개하면 다음과 같습니다.

$$\theta^{new} = \arg \max_{\theta} \mathbb{E}_{p_{\theta}(Z|X)} [\log p_{\theta^{old}}(Z) + \log p_{\theta^{old}}(X|Z)]$$

1. 2.

- 1. posterior $p_{\theta}(z|x)$ 에서 뽑은 sample이 prior $p_{\theta}(z)$ 에서 높은 likelihood를 가져야 한다.
2. posterior $p_{\theta}(z|x)$ 에서 뽑은 sample로 linear transform을 시킨 후 얻은 conditional distribution $p_{\theta}(x|z)$ 에서 가지고 있는 dataset이 높은 likelihood를 가져야 한다.
- 이는 곧 GMM에서 했던 EM 알고리즘의 M-step과 같은 식이 됩니다.

$$\theta^{new} = \arg \max_{\theta} \mathbb{E}_{p_{\theta}(Z|X)} [\log p_{\theta^{old}}(X, Z)]$$

Probabilistic PCA Expectation-Maximization

- M-step에서는 posterior에 의해 가상으로 만들어진 complete data의 log-likelihood를 최대화 합니다. complete data log-likelihood는 다음과 같이 계산됩니다.

- $\log p_\theta(X, Z) = \log p_\theta(Z) + \log p_\theta(X|Z)$

$$\begin{aligned} &= \log \prod_{n=1}^N p_\theta(z_n) + \log \prod_{n=1}^N p_\theta(x_n | z_n) \\ &= \sum_{n=1}^N \log p_\theta(z_n) + \sum_{n=1}^N \log p_\theta(x_n | z_n) \\ &= \sum_{n=1}^N \log N(z_n | 0, I) + \sum_{n=1}^N \log N(x_n | Wz_n + \mu, \sigma^2 I) \end{aligned}$$

- 또한 posterior $p_\theta(Z|X)$ 에 대한 complete data log-likelihood $\log p_\theta(X, Z)$ 의 기댓값은 다음과 같습니다.

$$\begin{aligned} \mathbb{E}_{p_\theta(Z|X)} [\ln p_\theta(\mathbf{X}, \mathbf{Z})] &= - \sum_{n=1}^N \left\{ \frac{D}{2} \ln (2\pi\sigma^2) + \frac{1}{2} \text{Tr} \left(\mathbb{E} [\mathbf{z}_n \mathbf{z}_n^\top] \right) \right. \\ &\quad + \frac{1}{2\sigma^2} \| \mathbf{x}_n - \boldsymbol{\mu} \|^2 - \frac{1}{\sigma^2} \mathbb{E} [\mathbf{z}_n]^\top \mathbf{W}^\top (\mathbf{x}_n - \boldsymbol{\mu}) \\ &\quad \left. + \frac{1}{2\sigma^2} \text{Tr} \left(\mathbb{E} [\mathbf{z}_n \mathbf{z}_n^\top] \mathbf{W}^\top \mathbf{W} \right) \right\} \end{aligned}$$

Probabilistic PCA Expectation-Maximization

- E-step : 각 data point마다 posterior $p_\theta(z|x)$ 를 구합니다.

$$p_\theta(z|x) = N(z|M^{-1}W^T(x - \mu), \sigma^2 M^{-1})$$

where $M = W^T W + \sigma^2 I$

- M-step : posterior를 이용해 parameters를 업데이트 합니다. μ 는 sample mean이 자명하므로 closed form으로 구합니다.

$$W_{new} = \left[\sum_{n=1}^N (x_n - \bar{x}) E[z_n]^T \right] \left[\sum_{n=1}^N E[z_n z_n^T] \right]^{-1}$$

$$\sigma_{new}^2 = \frac{1}{ND} \sum_{n=1}^N \left\{ \|x_n - \bar{x}\|^2 - 2E[z_n]^T W_{new}^T (x_n - \bar{x}) + Tr(E[z_n z_n^T] W_{new}^T W_{new}) \right\}$$

where $E[z_n z_n^T] = \sigma^2 M^{-1} + E[z_n] E[z_n]^T$

Bayesian PCA

Bayesian PCA

- PPCA(Probabilistic PCA)에서 사용했던 parameter W 를 random variable로 여기고 probability distribution을 갖도록 모델링 합니다.
- W 의 각 column마다 mean 0과 precision α_i 를 갖는 independent한 Gaussian distribution을 사용합니다.

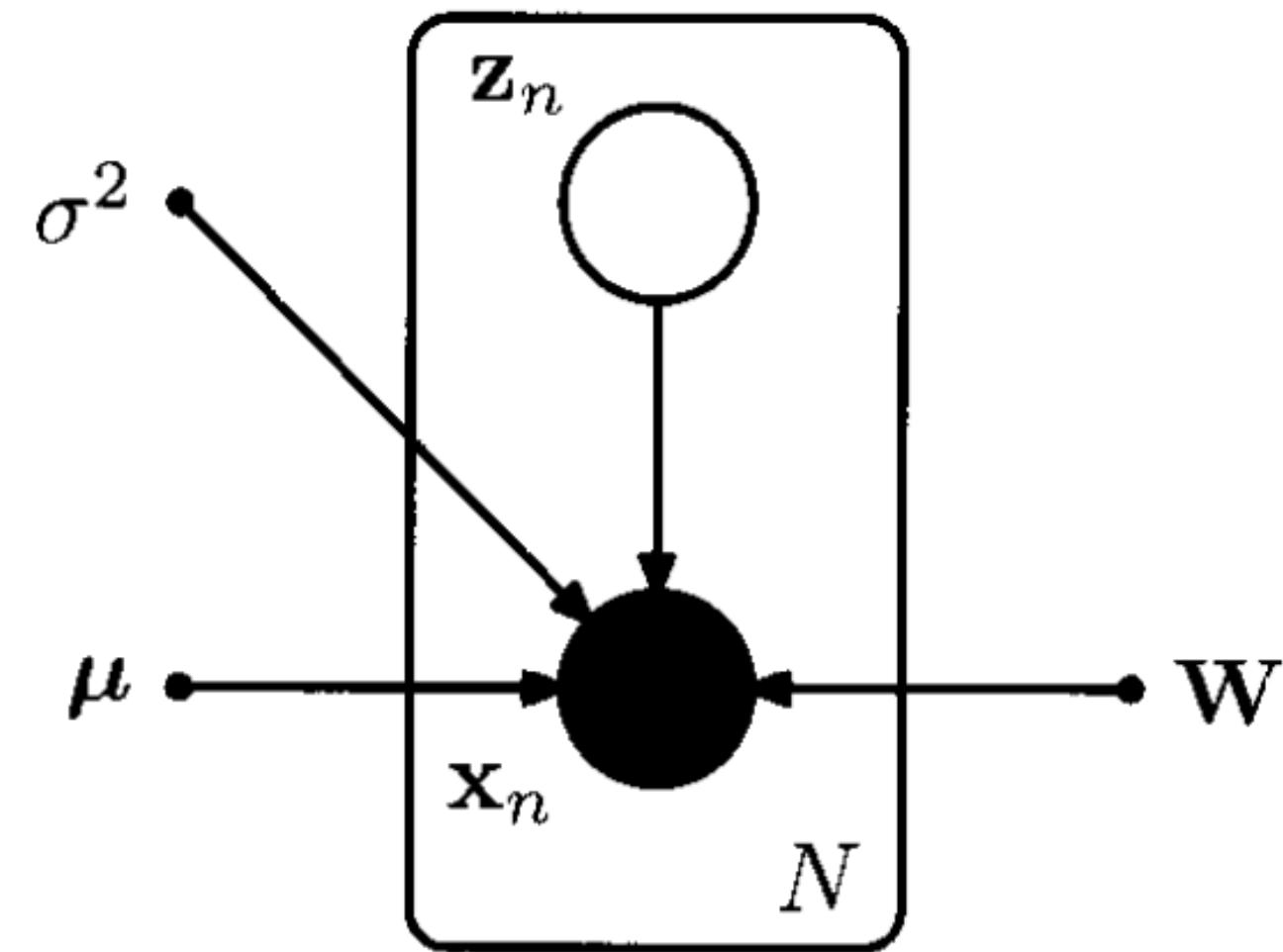
●

$$p(\mathbf{W} | \boldsymbol{\alpha}) = \prod_{i=1}^M \left(\frac{\alpha_i}{2\pi} \right)^{D/2} \exp \left\{ -\frac{1}{2} \alpha_i \mathbf{w}_i^T \mathbf{w}_i \right\}$$

$$\mathbf{W} = \begin{pmatrix} \vdots & \vdots & \vdots \\ N(\mathbf{w}_1 | 0, \sigma_1^{-1} I) & N(\mathbf{w}_2 | 0, \sigma_2^{-1} I) & \cdots & N(\mathbf{w}_M | 0, \sigma_M^{-1} I) \\ \vdots & \vdots & & \vdots \end{pmatrix}$$

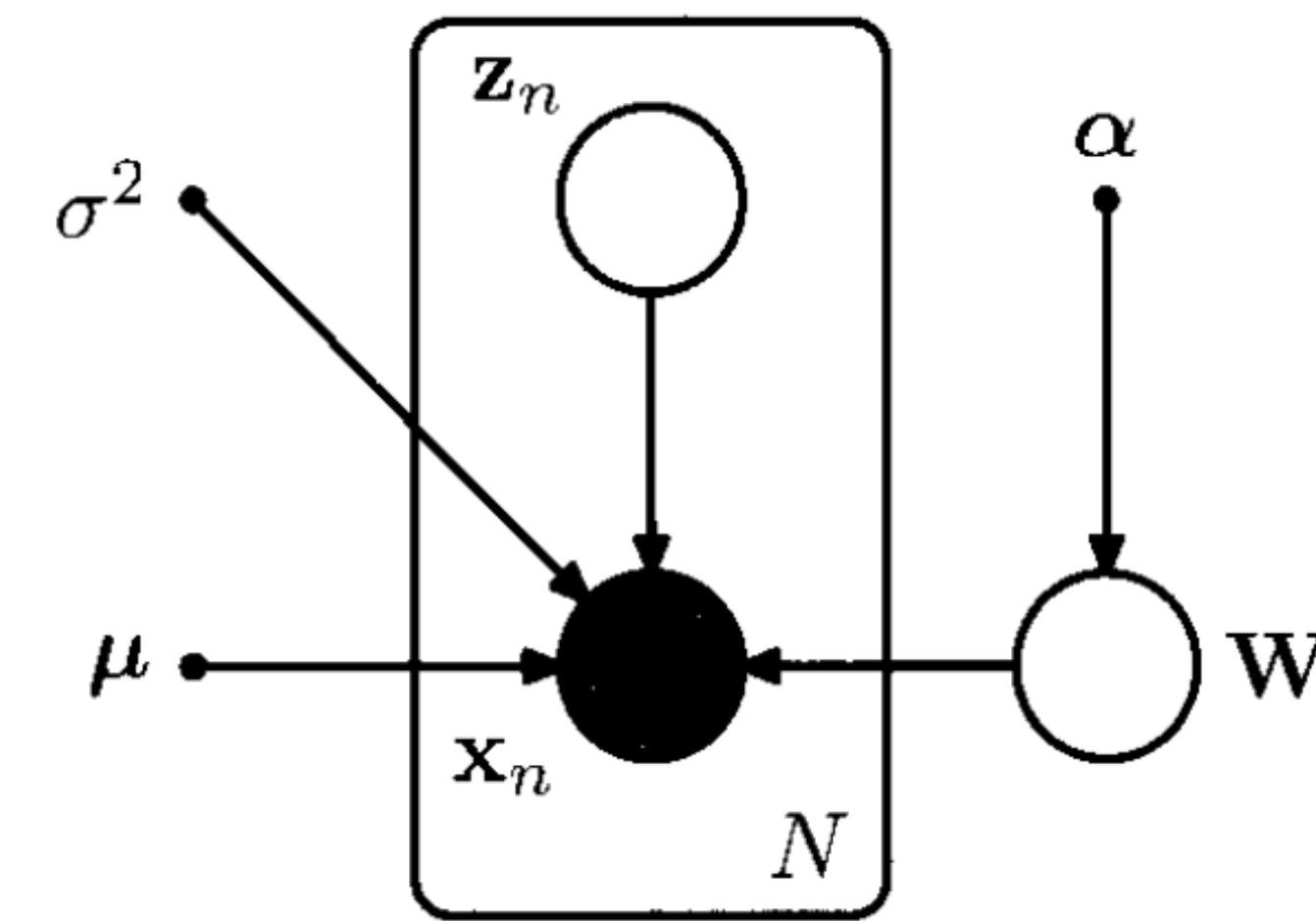
Bayesian PCA

Probabilistic PCA



Conditional $p_\theta(x | z) = N(x | Wz + \mu, \sigma^2 I)$

Bayesian PCA



$p_\theta(x | z, W) = N(x | Wz + \mu, \sigma^2 I)$

prior over W $p(W | \alpha) = \prod_{i=1}^M \left(\frac{\alpha_i}{2\pi} \right)^{D/2} \exp \left\{ -\frac{1}{2} \alpha_i \mathbf{w}_i^\top \mathbf{w}_i \right\}$

Bayesian PCA

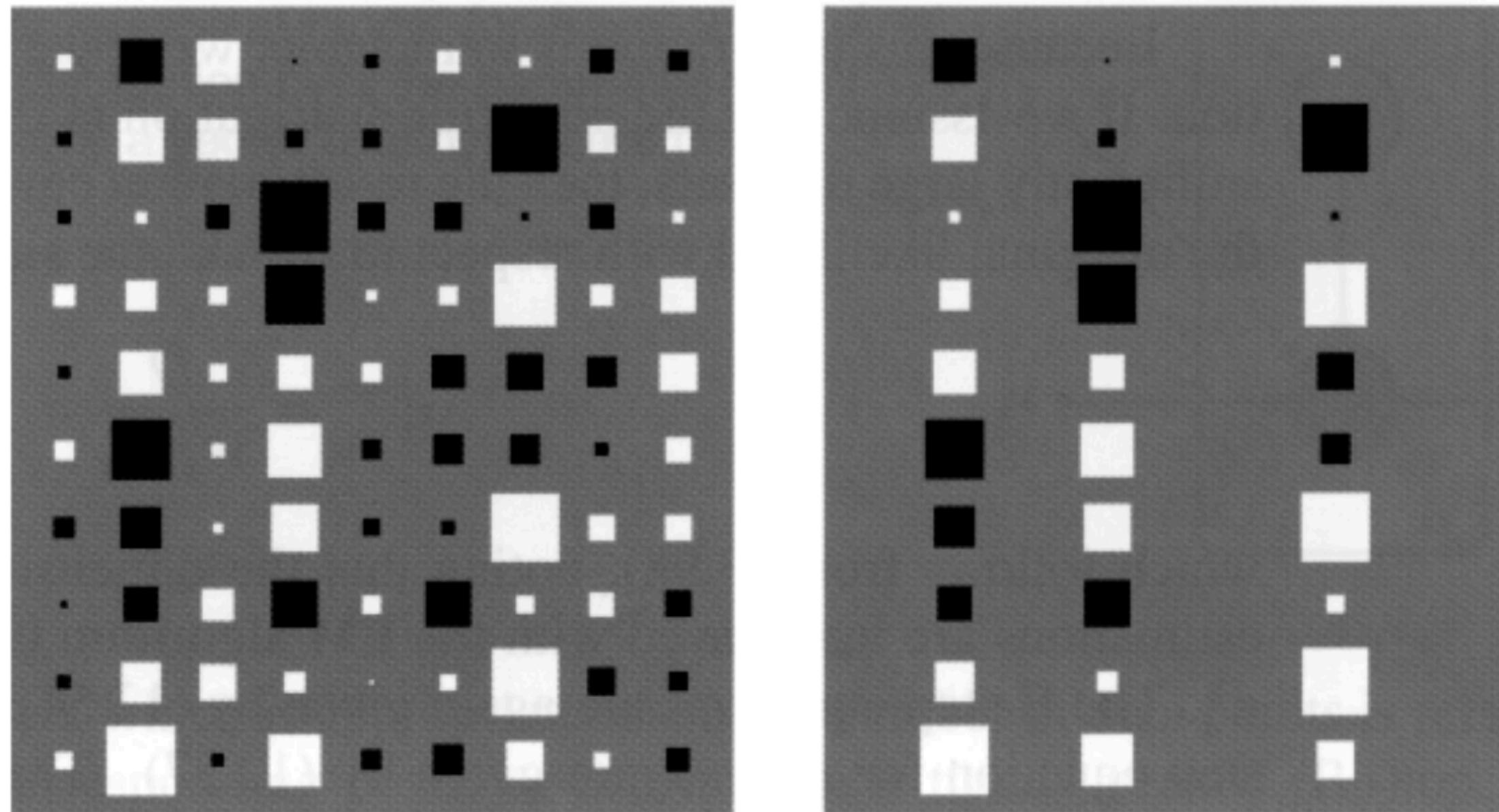


Figure 12.14 ‘Hinton’ diagrams of the matrix \mathbf{W} in which each element of the matrix is depicted as a square (white for positive and black for negative values) whose area is proportional to the magnitude of that element. The synthetic data set comprises 300 data points in $D = 10$ dimensions sampled from a Gaussian distribution having standard deviation 1.0 in 3 directions and standard deviation 0.5 in the remaining 7 directions for a data set in $D = 10$ dimensions having $M = 3$ directions with larger variance than the remaining 7 directions. The left-hand plot shows the result from maximum likelihood probabilistic PCA, and the left-hand plot shows the corresponding result from Bayesian PCA. We see how the Bayesian model is able to discover the appropriate dimensionality by suppressing the 6 surplus degrees of freedom.

Expectation & Maximization

Expectation & Maximization

- 앞서 GMM과 PPCA에서 해왔던 EM을 일반화한 형태를 써보면 다음과 같습니다.

- 1. Parameter θ_{old} 를 초기화 한다.

- 2. E-step : posterior $p(\mathbf{Z} | \mathbf{X}, \theta^{old})$ 를 구한다.

- 3. M-step : 새로운 parameter θ^{new} 를 구한다.

-

$$\theta^{new} = \arg \max_{\theta} Q(\theta, \theta^{old})$$

$$\text{where } Q(\theta, \theta^{old}) = \mathbb{E}_{p(\mathbf{Z} | \mathbf{X}, \theta^{old})}[p(\mathbf{X}, \mathbf{Z} | \theta)]$$

$$= \sum_{\mathbf{Z}} p(\mathbf{Z} | \mathbf{X}, \theta^{old}) \ln p(\mathbf{X}, \mathbf{Z} | \theta)$$

- 4. Log likelihood나 parameter 값이 수렴할 때까지 θ^{old} 를 θ^{new} 로 대체하고 E-step과 M-step을 반복한다.

KL-Divergence

Information Theory

- 다음과 같은 기준들에 의해 정보(information)을 수량화 합니다.
- 1. 자주 일어나는 사건은 낮은 정보량을 갖는다.
 2. 드물게 일어나는 사건은 높은 정보량을 갖는다.
 3. 독립된 사건의 정보량은 각 사건의 정보량을 더하여 구한다.
- $$h(x) = -\log p(x)$$
- Random variable x 의 distribution이 $p(x)$ 일 때, x 의 엔트로피(entropy)는 다음과 같이 정보량의 기댓값으로 정의합니다.
- $$H[x] = \mathbb{E}_{p(x)}[h(x)] = \begin{cases} \sum_x p(x)\{-\log p(x)\}, & \text{discrete} \\ \int p(x)\{-\log p(x)\}dx, & \text{continuous} \end{cases}$$

KL-Divergence

- KL-divergence는 두 분포의 차이를 재는 범함수(functional)입니다.

- 두 분포 P, Q 에 대해서 다음과 같이 KL-divergence를 정의합니다.

$$D_{\text{KL}}(P\|Q) = \mathbb{E}_{x \sim P} \left[\log \frac{P(x)}{Q(x)} \right] = \begin{cases} \sum_x P(x) \log \frac{P(x)}{Q(x)}, & \text{discrete} \\ \int P(x) \log \frac{P(x)}{Q(x)} dx, & \text{continuous} \end{cases}$$

- KL-divergence는 symmetric하지 않으므로 distance의 개념이 아닙니다.

-

$$D_{\text{KL}}(P\|Q) \neq D_{\text{KL}}(Q\|P)$$

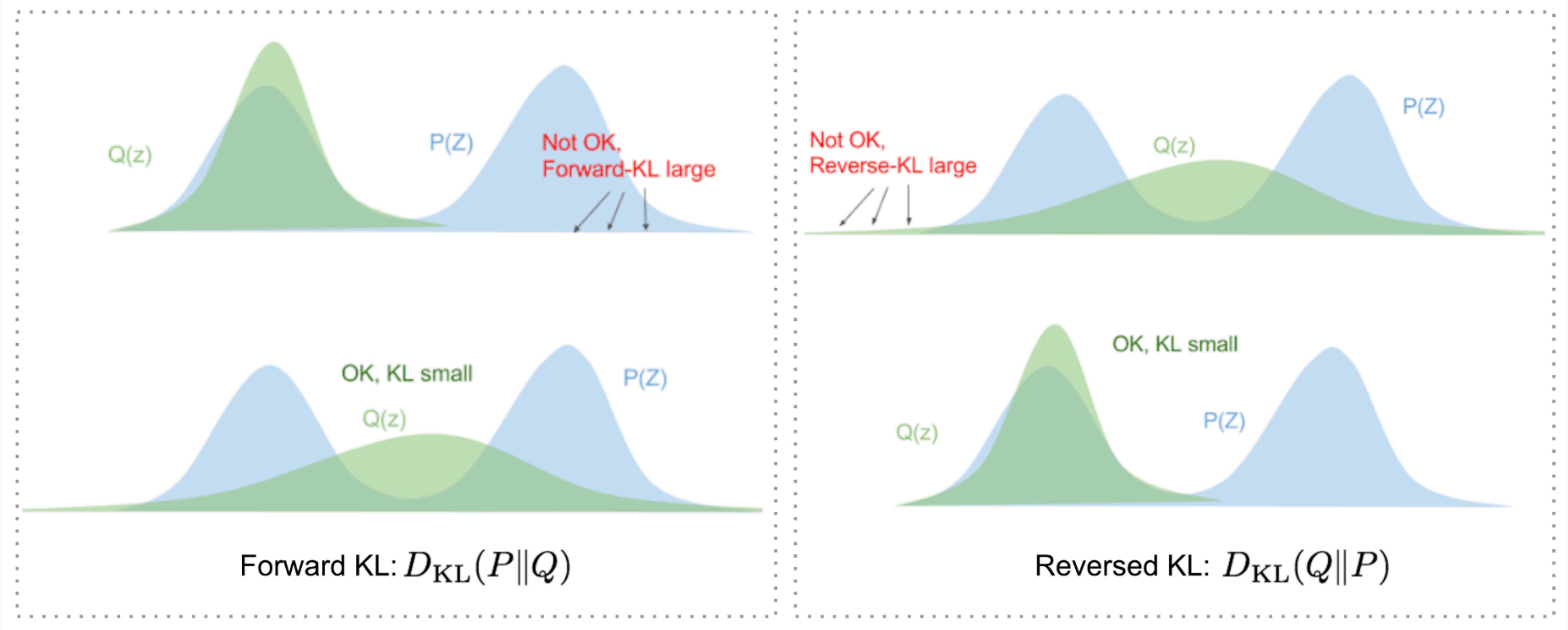
- KL-divergence 값은 항상 0보다 같거나 크며, 두 distribution P, Q 가 같을 때만 0이 됩니다.

-

$$\mathbb{E}_{x \sim P} \left[-\log \frac{Q(x)}{P(x)} \right] \geq -\log \left(\mathbb{E}_{x \sim P} \left[\frac{Q(x)}{P(x)} \right] \right) = -\log \left(\sum_x P(x) \frac{Q(x)}{P(x)} \right) = 0$$

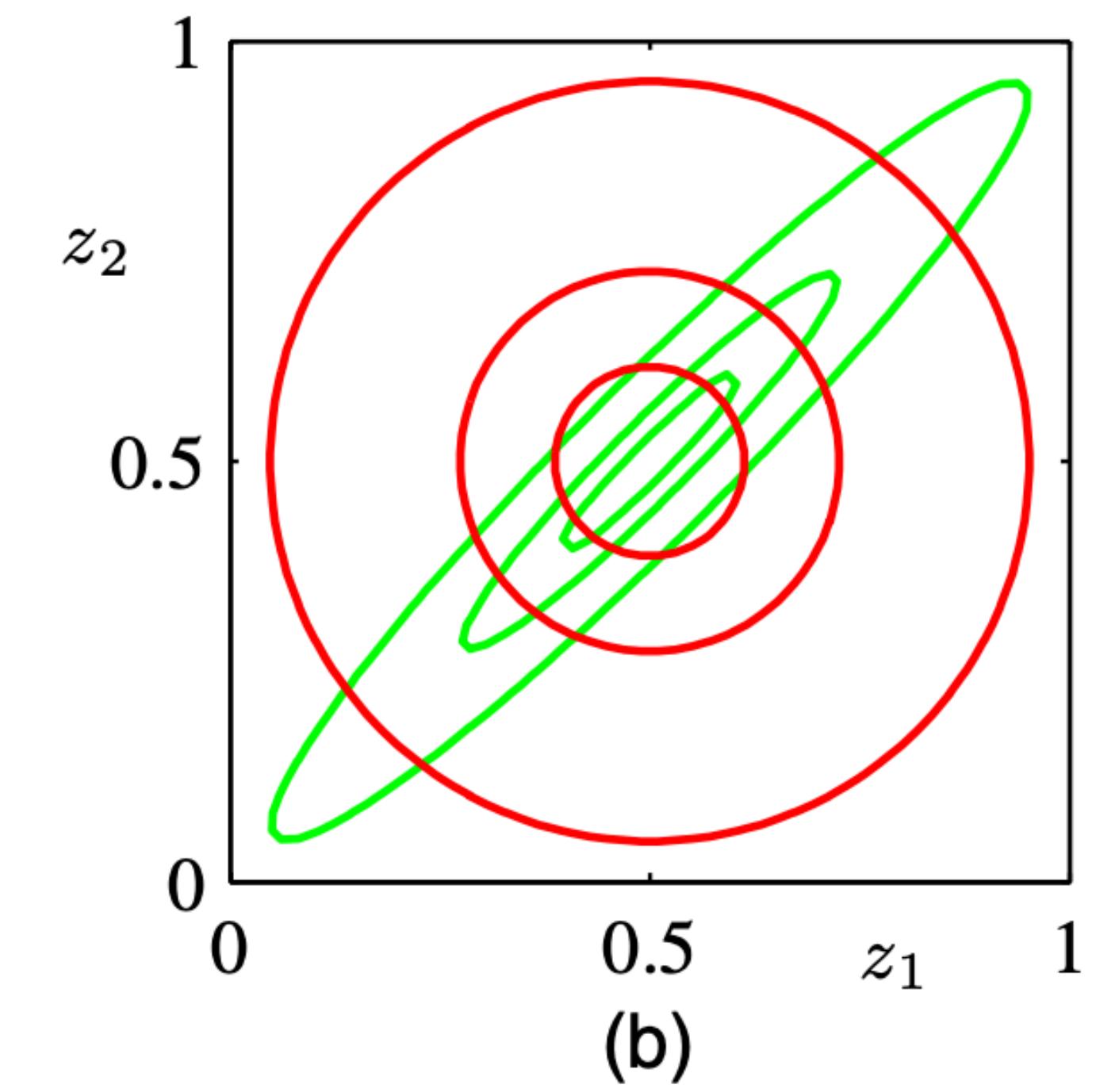
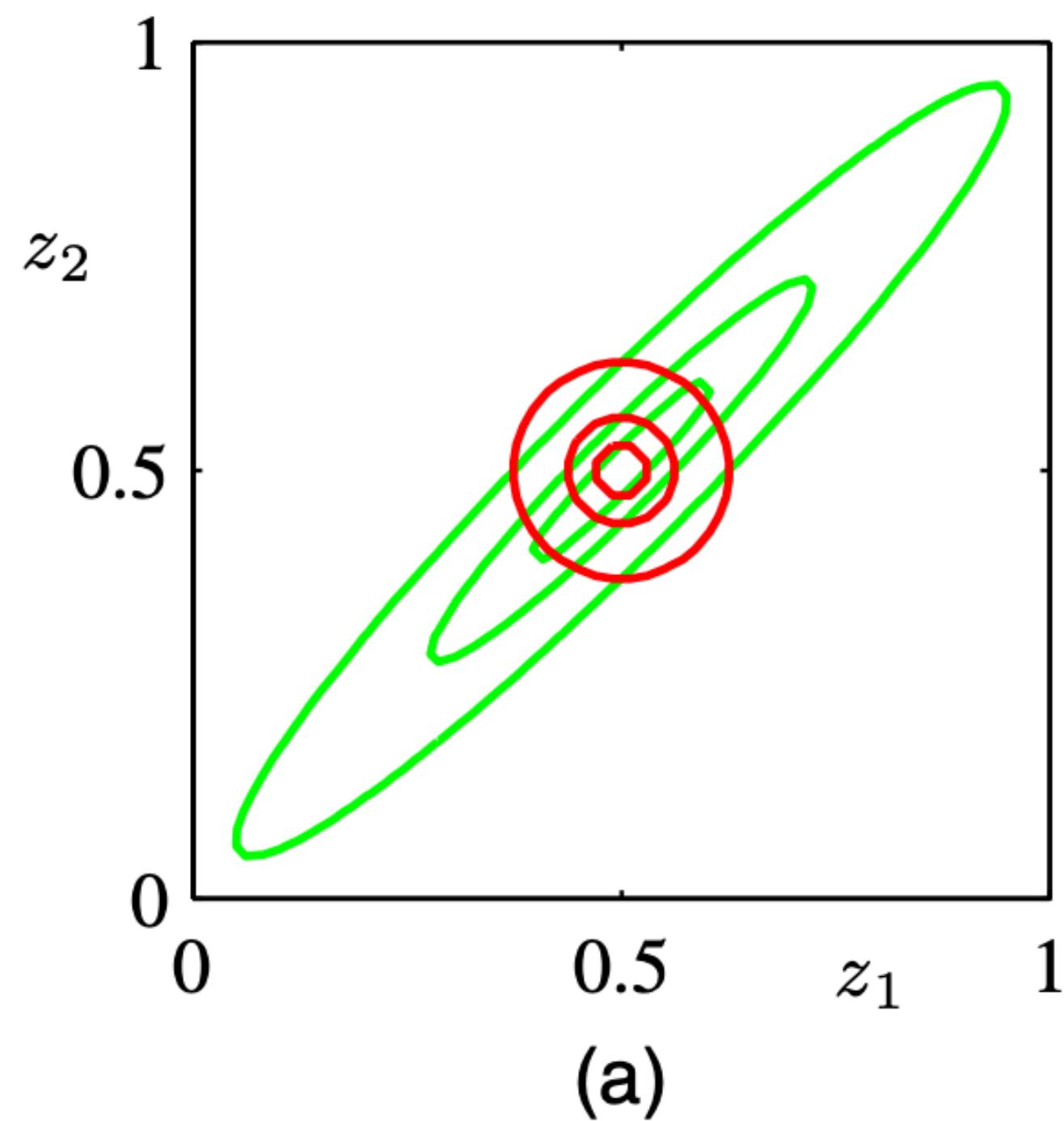
by Jensen's inequality

KL-Divergence



KL-Divergence

Figure 10.2 Comparison of the two alternative forms for the Kullback-Leibler divergence. The green contours corresponding to 1, 2, and 3 standard deviations for a correlated Gaussian distribution $p(\mathbf{z})$ over two variables z_1 and z_2 , and the red contours represent the corresponding levels for an approximating distribution $q(\mathbf{z})$ over the same variables given by the product of two independent univariate Gaussian distributions whose parameters are obtained by minimization of (a) the Kullback-Leibler divergence $\text{KL}(q||p)$, and (b) the reverse Kullback-Leibler divergence $\text{KL}(p||q)$.



KL-Divergence

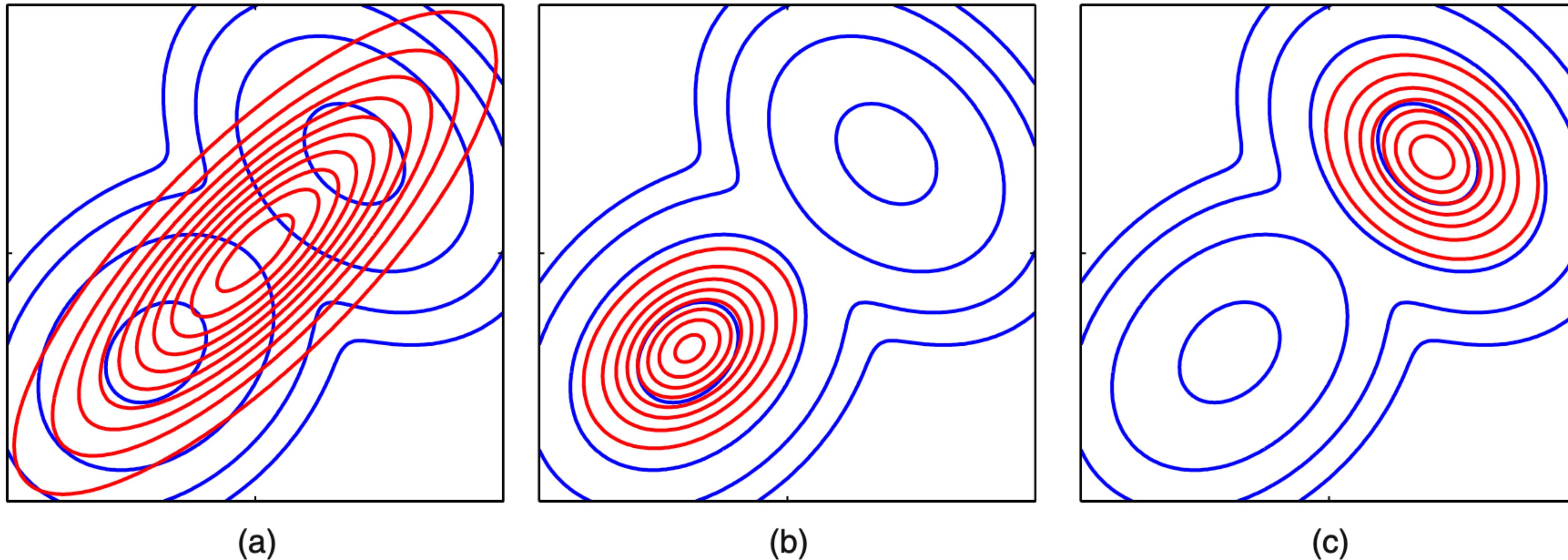


Figure 10.3 Another comparison of the two alternative forms for the Kullback-Leibler divergence. (a) The blue contours show a bimodal distribution $p(\mathbf{Z})$ given by a mixture of two Gaussians, and the red contours correspond to the single Gaussian distribution $q(\mathbf{Z})$ that best approximates $p(\mathbf{Z})$ in the sense of minimizing the Kullback-Leibler divergence $\text{KL}(p\|q)$. (b) As in (a) but now the red contours correspond to a Gaussian distribution $q(\mathbf{Z})$ found by numerical minimization of the Kullback-Leibler divergence $\text{KL}(q\|p)$. (c) As in (b) but showing a different local minimum of the Kullback-Leibler divergence.

General EM

General EM-posterior 계산의 문제점

- EM 알고리즘은 E-step에서 posterior $p_\theta(\mathbf{Z} | \mathbf{X})$ 를 구하는 과정이 필수적입니다.

$$p_\theta(\mathbf{Z} | \mathbf{X}) = \frac{p_\theta(\mathbf{Z})p_\theta(\mathbf{X} | \mathbf{Z})}{p_\theta(\mathbf{X})}$$

- 그러나 분모에 있는 $p_\theta(\mathbf{X})$ 를 구하기 위해 적분 또는 summation을 해야 하는데 실제로 어려운 (intractable) 경우가 있습니다.

$$\begin{aligned} p_\theta(\mathbf{X}) &= \int p_\theta(\mathbf{Z})p_\theta(\mathbf{X} | \mathbf{Z})d\mathbf{Z}, \mathbf{Z} \text{가 continuous인 경우,} \\ &= \sum_{\mathbf{Z}} p_\theta(\mathbf{Z})p_\theta(\mathbf{X} | \mathbf{Z}), \mathbf{Z} \text{가 discrete인 경우} \end{aligned}$$

General EM-Variational Distribution의 도입

- E-step에서 posterior $p_{\theta}(\mathbf{Z} | \mathbf{X})$ 를 구하는 것은 다음과 같이 KL-divergence를 최소화 하는 variational distribution $q_{\phi}(\mathbf{Z})$ 를 구하는 것으로 대체할 수 있습니다.

$$q_{\phi}(\mathbf{Z}) = \arg \min_{q_{\phi}(\mathbf{Z})} \text{KL}(q || p)$$

where $\text{KL}(q || p) = - \mathbb{E}_{q_{\phi}(\mathbf{Z})} [\ln p_{\theta}(\mathbf{Z} | \mathbf{X}) - q_{\phi}(\mathbf{Z})]$

$$= - \sum_{\mathbf{Z}} q_{\phi}(\mathbf{Z}) \ln \left\{ \frac{p_{\theta}(\mathbf{Z} | \mathbf{X})}{q_{\phi}(\mathbf{Z})} \right\}$$

- $\text{KL}(q || p)$ 은 $q_{\phi}(\mathbf{Z})$ 와 $p_{\theta}(\mathbf{Z} | \mathbf{X})$ 가 같을 때 0이 되므로 $p_{\theta}(\mathbf{Z} | \mathbf{X})$ 를 대신해 $q_{\phi}(\mathbf{Z})$ 를 사용할 수 있습니다.

General EM-ELBO 사용

- posterior $p_{\theta}(\mathbf{Z} | \mathbf{X})$ 를 구하기 어려운 경우 $\text{KL}(q||p)$ 식을 직접 계산할 수 없으므로 우회하여 구해봅니다.
- Marginal log-likelihood $\ln p_{\theta}(\mathbf{X})$ 와 ELBO라 불리는 $\mathcal{L}(q, \theta)$ 와 KL-divergence $\text{KL}(q||p)$ 간에는 다음과 같은 식이 성립합니다.
-

$$\ln p_{\theta}(\mathbf{X}) = \mathcal{L}(q, \theta) + \text{KL}(q||p)$$

$$\text{where } \mathcal{L}(q, \theta) = \sum_{\mathbf{Z}} q_{\phi}(\mathbf{Z}) \ln \left\{ \frac{p_{\theta}(\mathbf{X}, \mathbf{Z})}{q_{\phi}(\mathbf{Z})} \right\}$$

$$\text{KL}(q||p) = - \sum_{\mathbf{Z}} q_{\phi}(\mathbf{Z}) \ln \left\{ \frac{p_{\theta}(\mathbf{Z} | \mathbf{X})}{q_{\phi}(\mathbf{Z})} \right\}$$

General EM-ELBO 사용

- Parameter θ 가 고정되어 있을 때 marginal log-likelihood $\ln p_\theta(\mathbf{X})$ 는 고정이므로 KL-divergence $\text{KL}(q\|p)$ 를 최소화하는 것은 ELBO $\mathcal{L}(q, \theta)$ 를 최대화하는 것이 됩니다.
- 이점을 이용해 이용해 $q_\phi(\mathbf{Z})$ 를 $\text{KL}(q\|p)$ 를 최소화하는 것이 아닌 $\mathcal{L}(q, \theta)$ 를 최대화하는 방향으로 $p_\theta(\mathbf{Z} | \mathbf{X})$ 를 근사할 수 있습니다.
-

$$q_\phi(\mathbf{Z}) = \arg \max_{q_\phi(\mathbf{Z})} \mathcal{L}(q, \theta)$$

General EM-ELBO 유도

-

$$\ln p_{\theta}(\mathbf{X}) = \sum_{\mathbf{Z}} q_{\phi}(\mathbf{Z}) \ln p_{\theta}(\mathbf{X})$$

$$= \sum_{\mathbf{Z}} q_{\phi}(\mathbf{Z}) \ln \frac{p_{\theta}(\mathbf{X}, \mathbf{Z})}{p_{\theta}(\mathbf{Z} | \mathbf{X})}$$

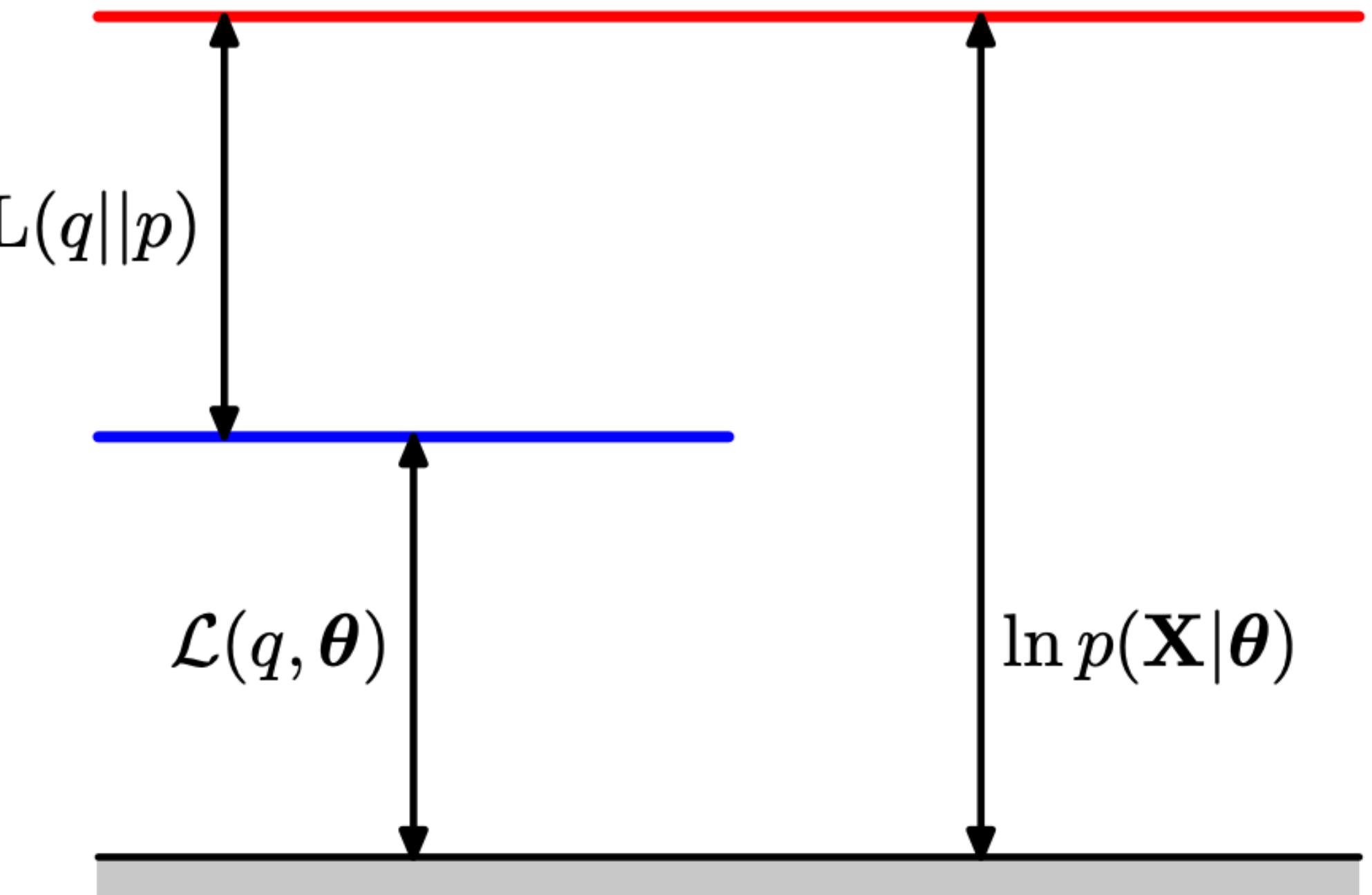
$$= \sum_{\mathbf{Z}} q_{\phi}(\mathbf{Z}) \ln \frac{p_{\theta}(\mathbf{X}, \mathbf{Z})}{q_{\phi}(\mathbf{Z})} - \frac{q_{\phi}(\mathbf{Z})}{p_{\theta}(\mathbf{Z} | \mathbf{X})}$$

$$= \sum_{\mathbf{Z}} q_{\phi}(\mathbf{Z}) \ln \frac{p_{\theta}(\mathbf{X}, \mathbf{Z})}{q_{\phi}(\mathbf{Z})} - \sum_{\mathbf{Z}} q_{\phi}(\mathbf{Z}) \ln \frac{p_{\theta}(\mathbf{Z} | \mathbf{X})}{q_{\phi}(\mathbf{Z})}$$

$$= \mathcal{L}(q, \theta) + \text{KL}(q \| p)$$

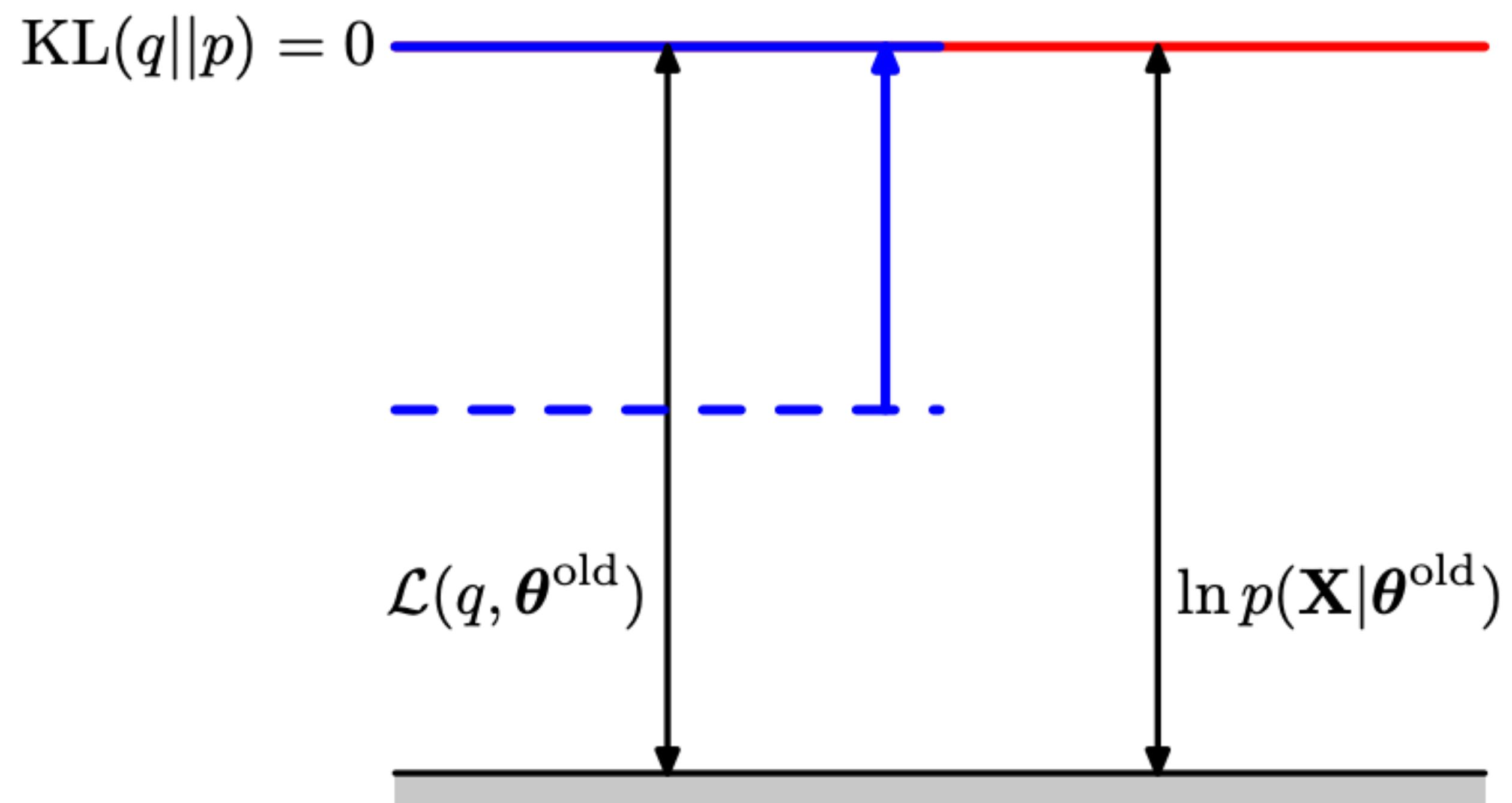
General EM

Figure 9.11 Illustration of the decomposition given by (9.70), which holds for any choice of distribution $q(\mathbf{Z})$. Because the Kullback-Leibler divergence satisfies $\text{KL}(q||p) \geq 0$, we see that the quantity $\mathcal{L}(q, \theta)$ is a lower bound on the log likelihood function $\ln p(\mathbf{X}|\theta)$.



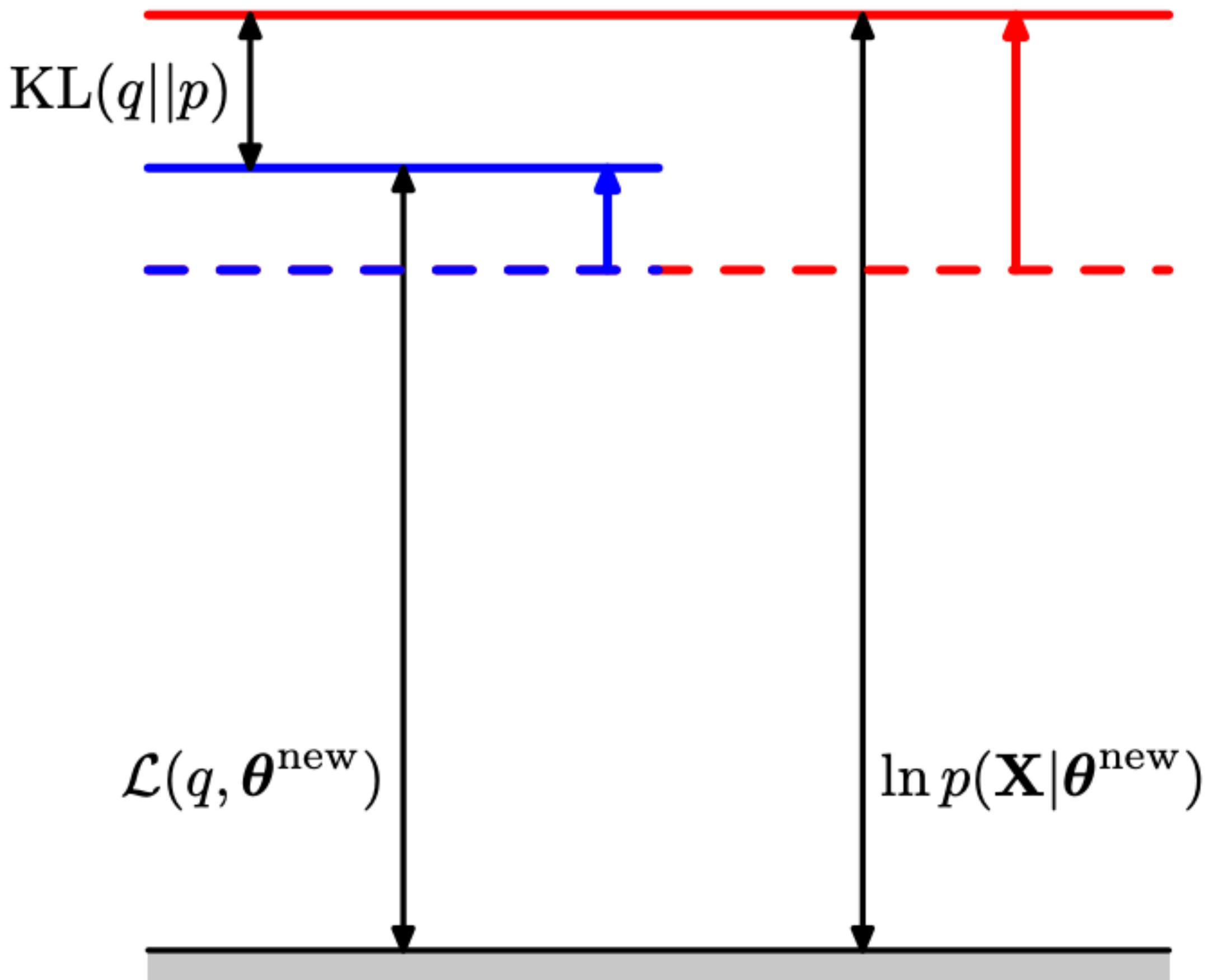
General EM

Figure 9.12 Illustration of the E step of the EM algorithm. The q distribution is set equal to the posterior distribution for the current parameter values θ^{old} , causing the lower bound to move up to the same value as the log likelihood function, with the KL divergence vanishing.



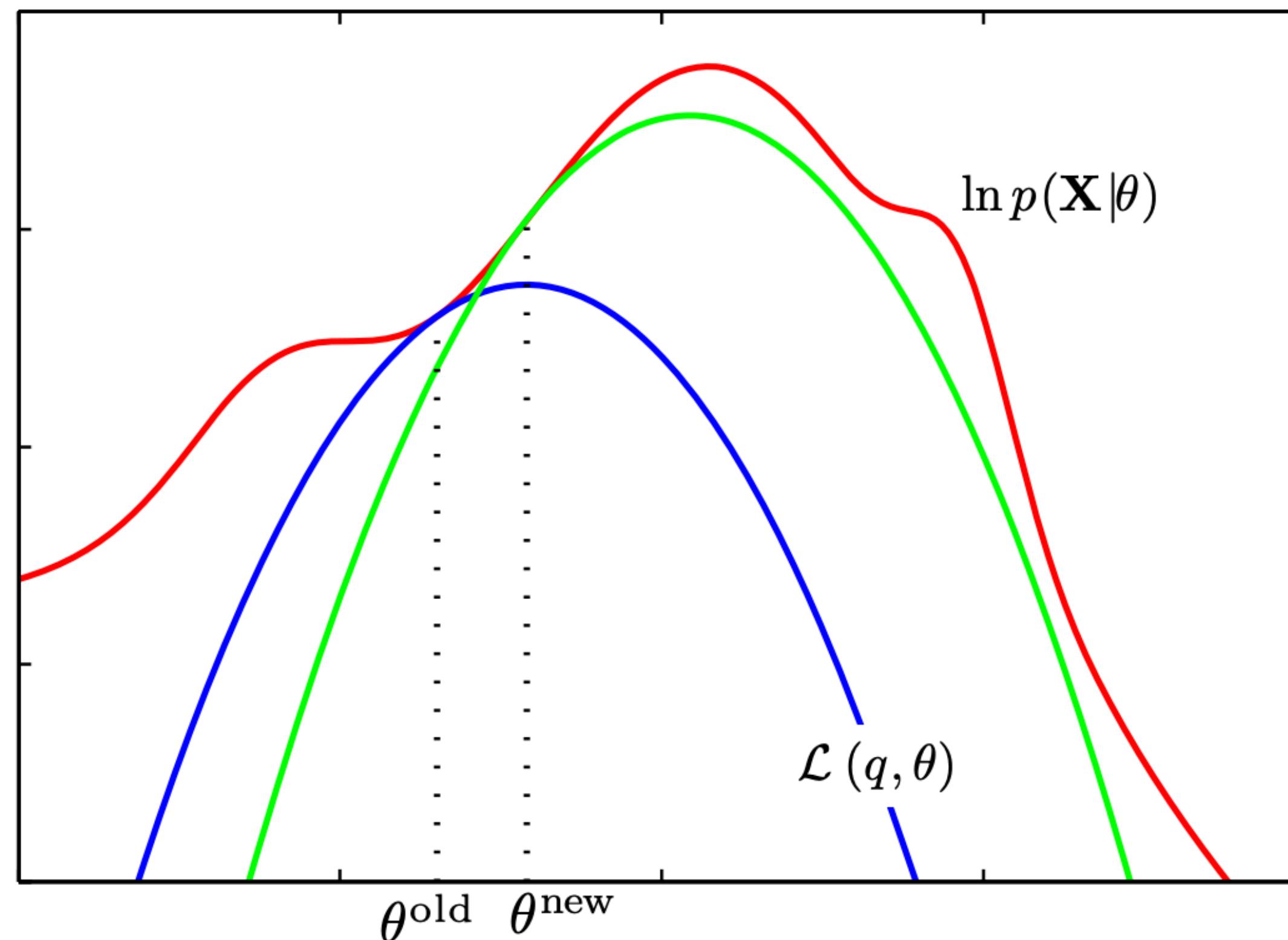
General EM

Figure 9.13 Illustration of the M step of the EM algorithm. The distribution $q(\mathbf{Z})$ is held fixed and the lower bound $\mathcal{L}(q, \theta)$ is maximized with respect to the parameter vector θ to give a revised value θ^{new} . Because the KL divergence is nonnegative, this causes the log likelihood $\ln p(\mathbf{X}|\theta)$ to increase by at least as much as the lower bound does.

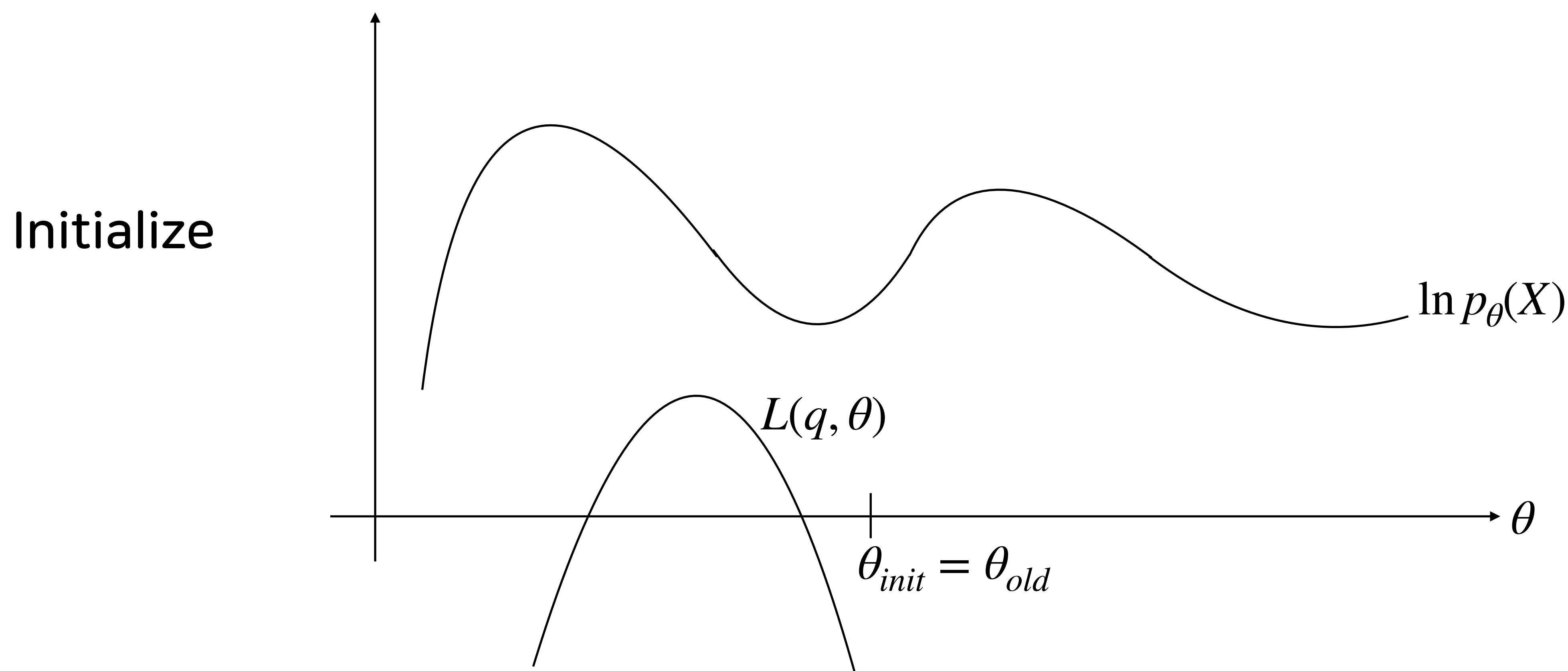


General EM

Figure 9.14 The EM algorithm involves alternately computing a lower bound on the log likelihood for the current parameter values and then maximizing this bound to obtain the new parameter values. See the text for a full discussion.

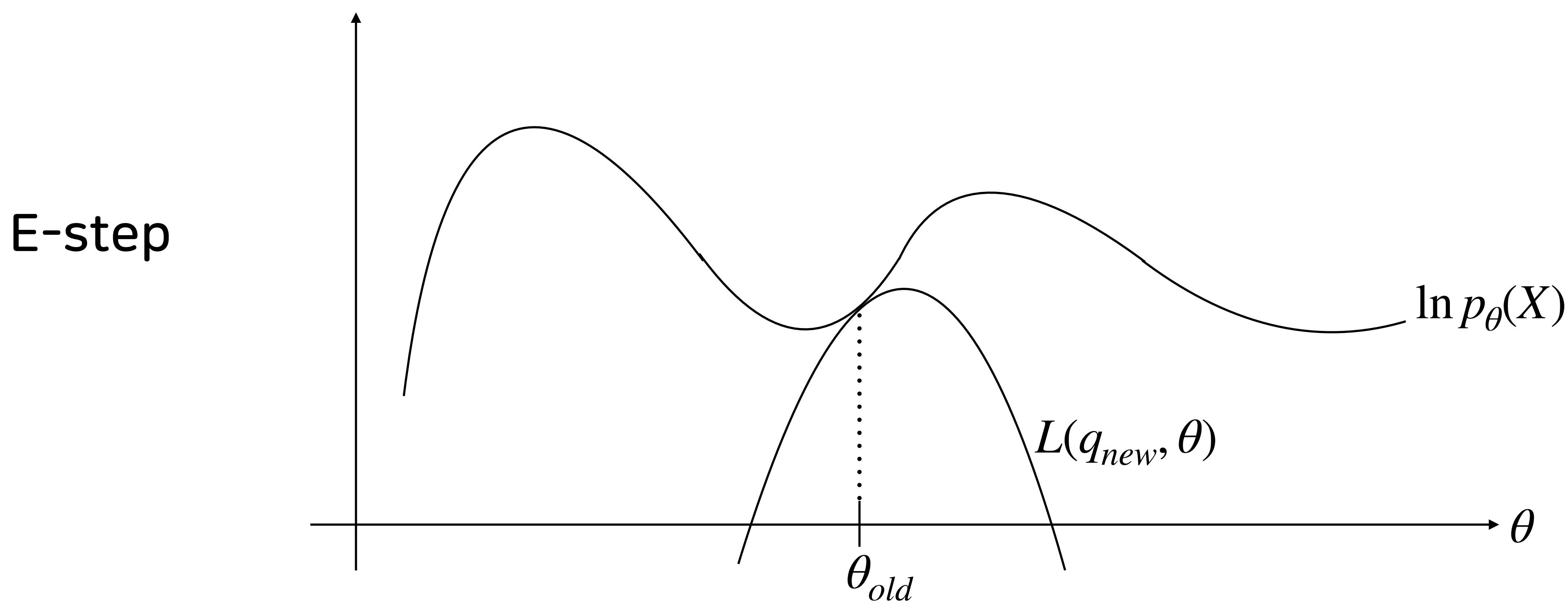


General EM



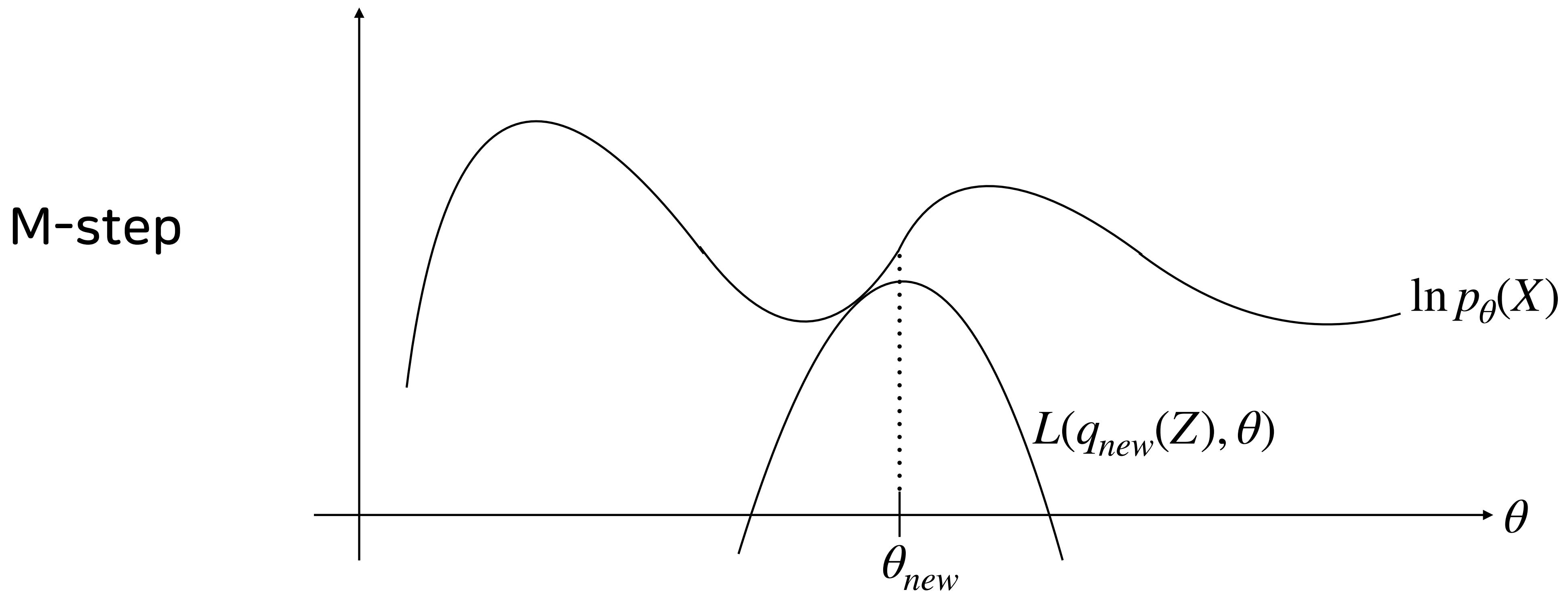
General EM

Set $q_{new}(Z) = p_{\theta_{old}}(Z|X)$



General EM

Find the θ_{new} that maximize $L(q_{new}(Z), \theta)$

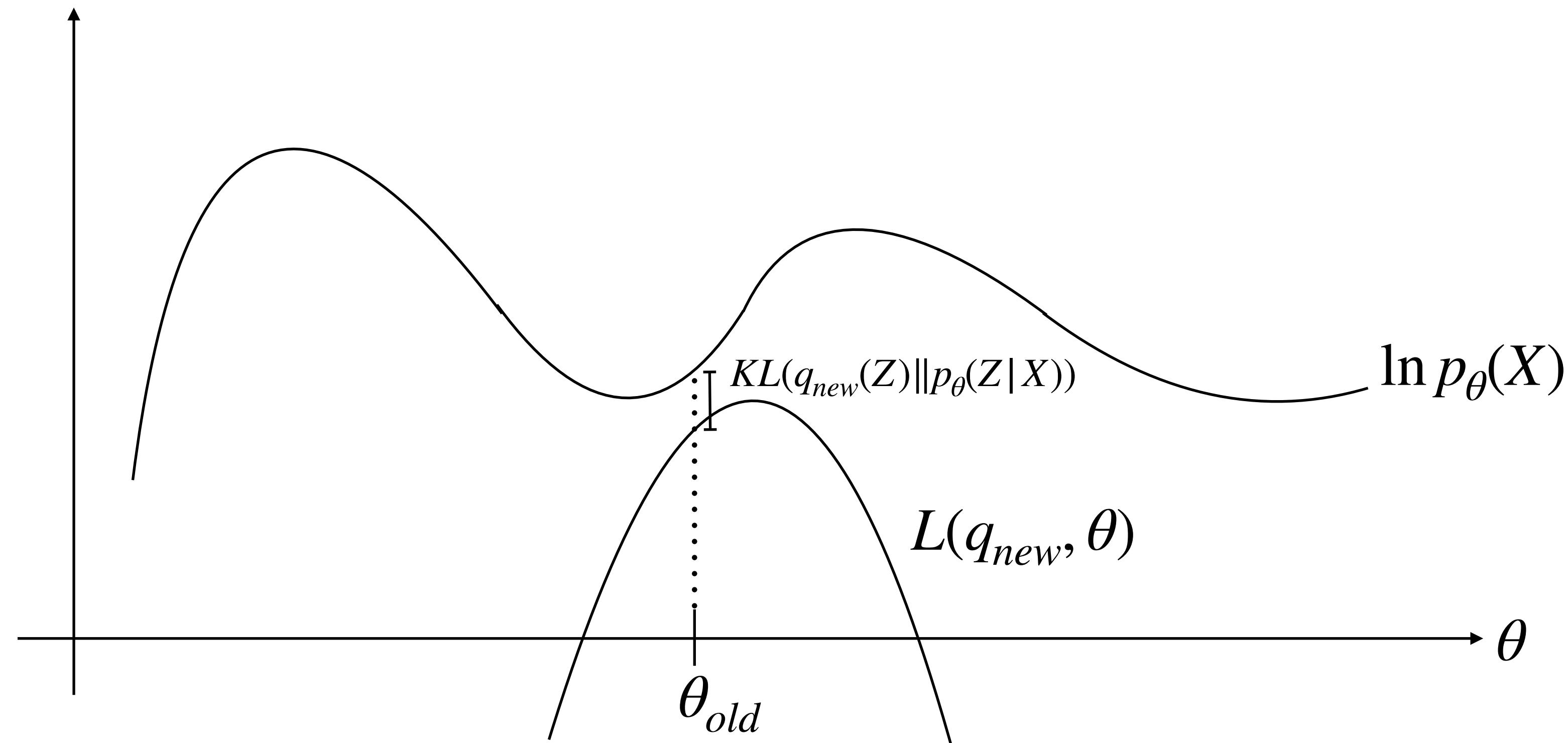


General EM

~~Set $q_{new}(Z) = p_{\theta_{old}}(Z|X)$~~

Find the $q_{new}(Z)$ that maximize $L(q_{new}, \theta)$

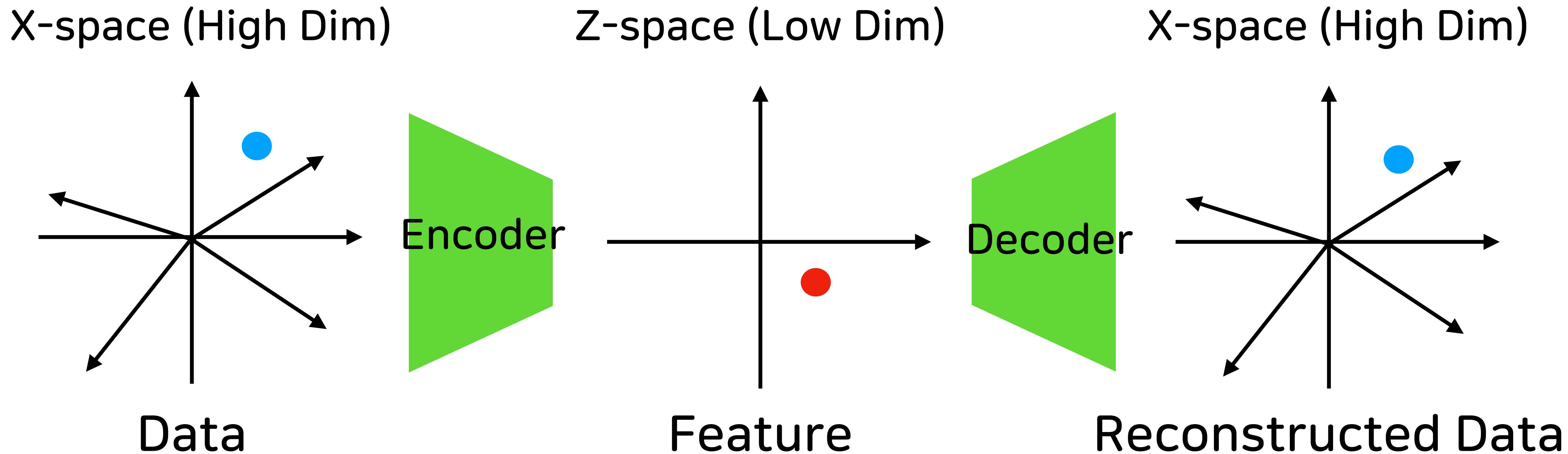
Variational
E-step



Auto-Encoder

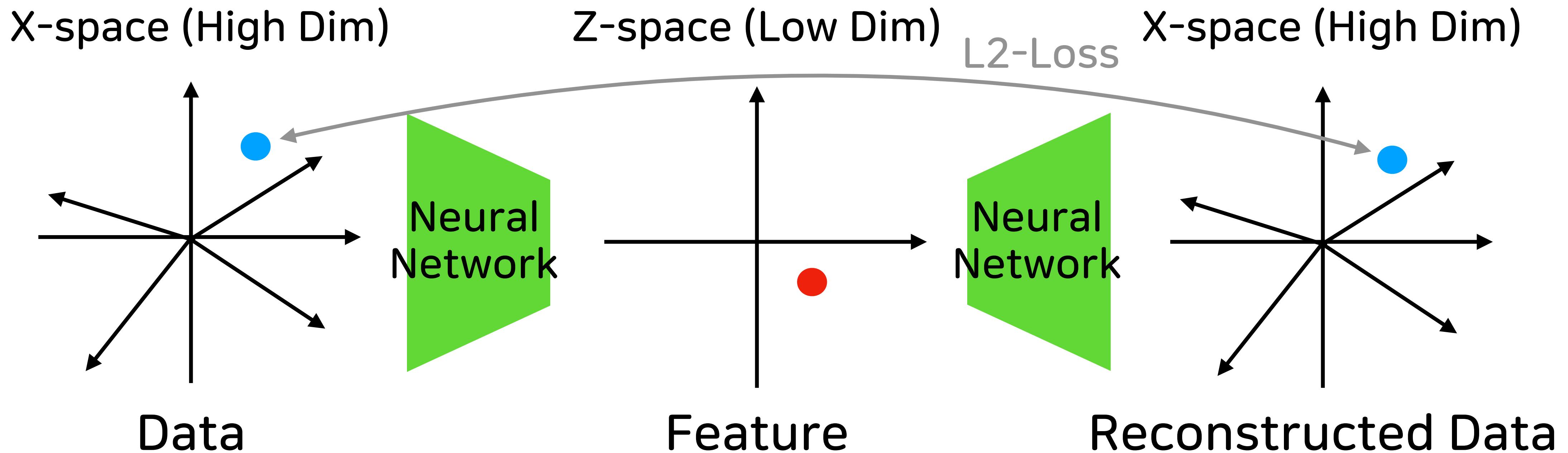
Auto-Encoder(개념)

- Auto-Encoder는 Encoder를 통해 고차원의 데이터를 저차원으로 맵핑하고, Decoder를 통해 다시 복원하는 네트워크입니다.
- 이렇게 저차원으로 맵핑된 정보는 데이터를 나타내는 feature로 활용될 수 있습니다.



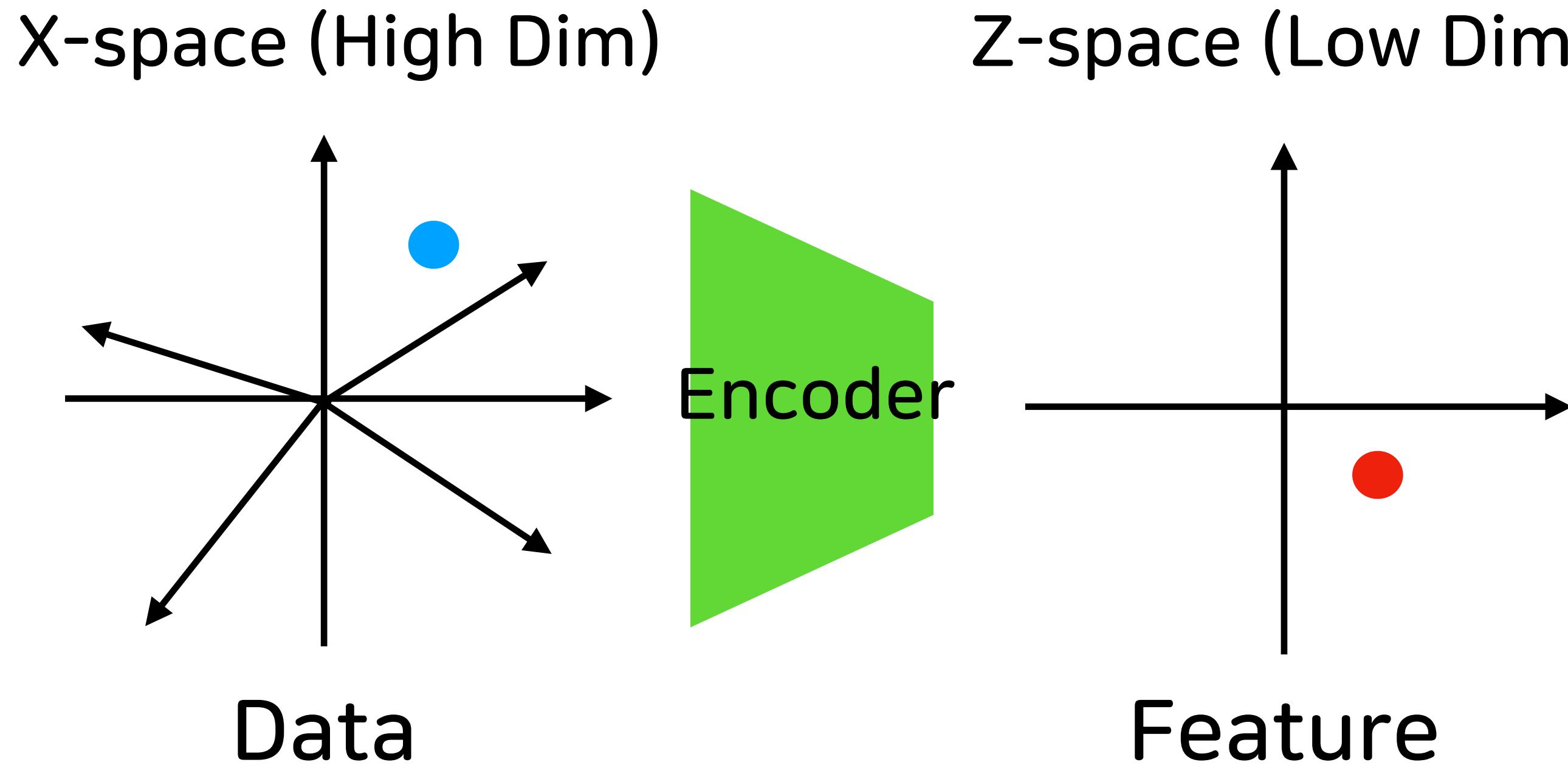
Auto-Encoder(구성)

- Encoder와 Decoder는 Fully-Connected Layers 또는 Convolution Layers 등 Neural Networks로 구성합니다.
- input과 output 간에 L2 Loss를 취해 reconstruction을 할 수 있도록 합니다.



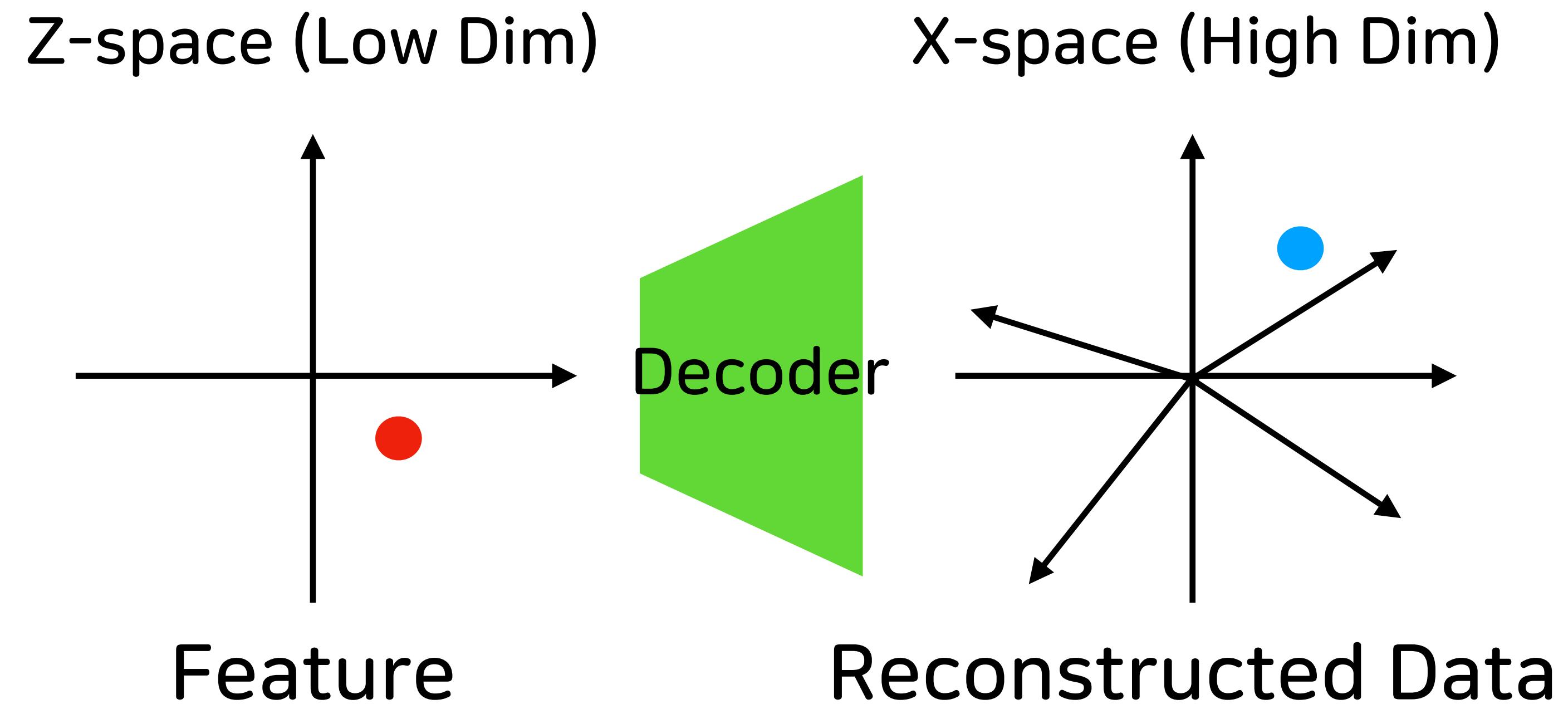
Auto-Encoder(Encoder)

- 트레이닝 후 Encoder는 Feature Extraction의 용도로 활용될 수 있습니다.
- 이 때, Encoder는 고차원의 데이터를 압축하여 feature를 추출해내는 역할을 합니다.

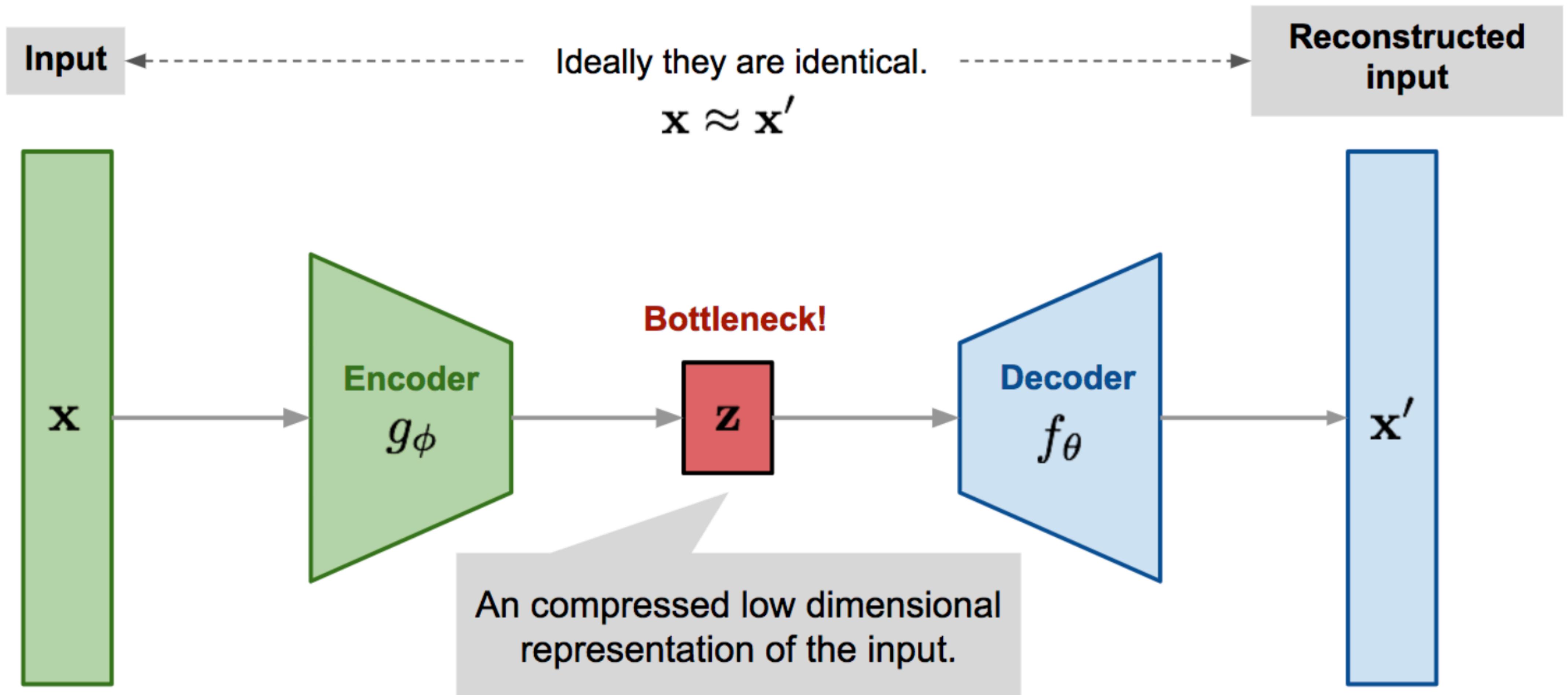


Auto-Encoder(Decoder)

- Z-space에서 벡터 하나를 샘플링하여 Decoder를 통해 X-space의 벡터로 복원하면 데이터를 생성해 낼 수 있습니다.
- 하지만 Z-space에서 벡터들의 집합이 어떤 분포를 가지고 있는지 Auto-Encoder 만으로는 알 수 없으므로 생성모델이라 부를 수 없습니다.



Auto-Encoder



Auto-Encoder

