Week7 – GMM & EM Algorithm

1부 - 오현도

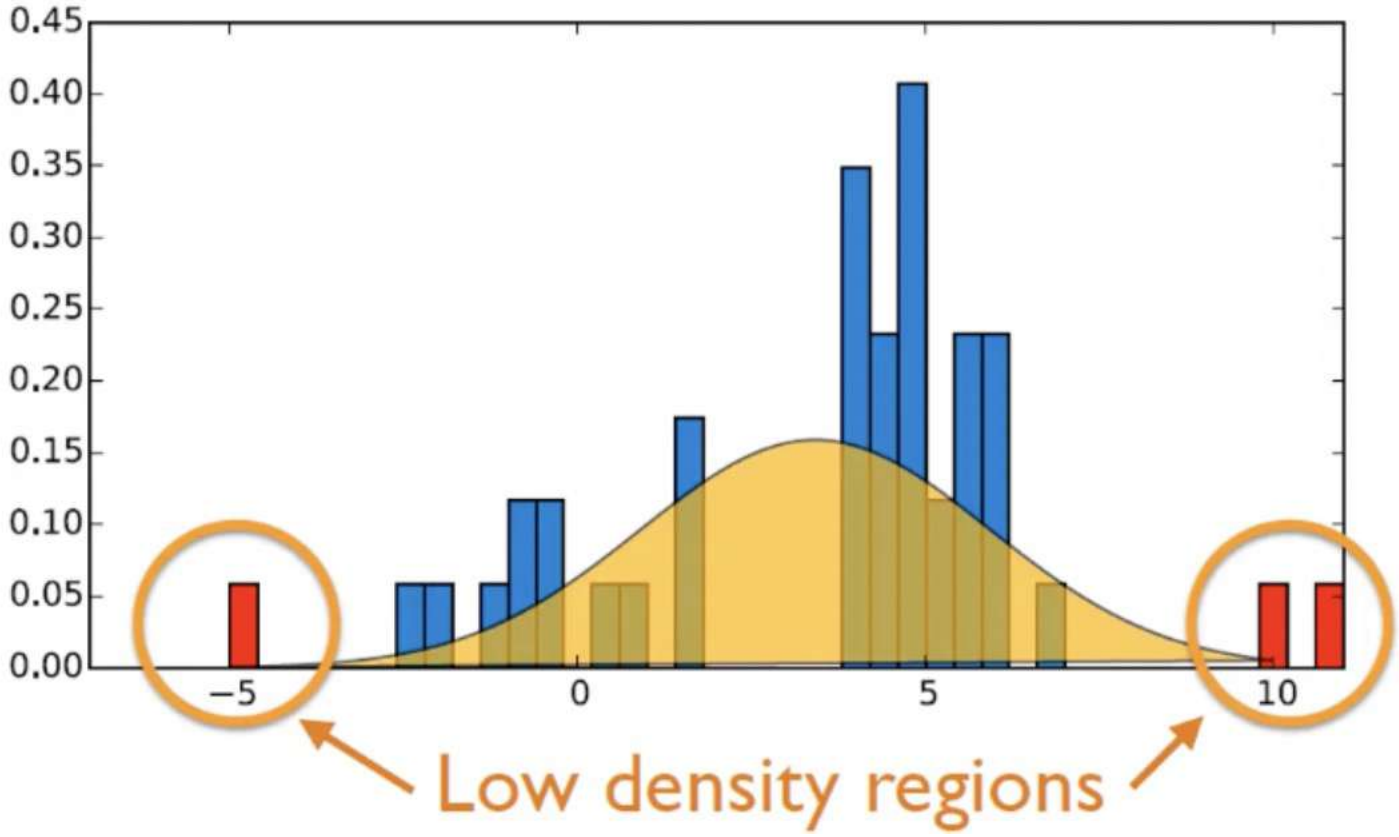
2부 - 이규민

Contents

Contents

Gaussian Density Estimation

Gaussian Density Estimation



Anomaly Detection..

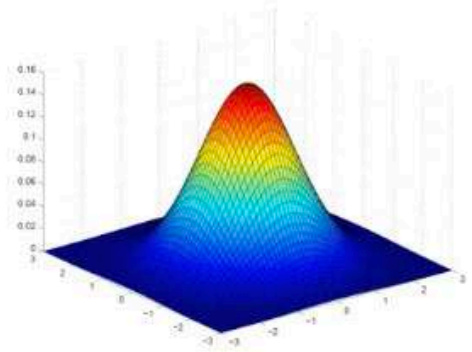
Data-driven density function을

Parameteric하게,

가우시안 분포로 가정을 하겠다.

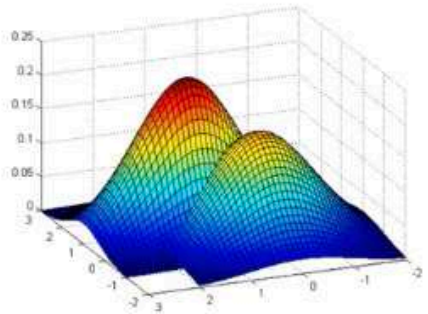
Gaussian Density Estimation

Gaussian Density Estimation



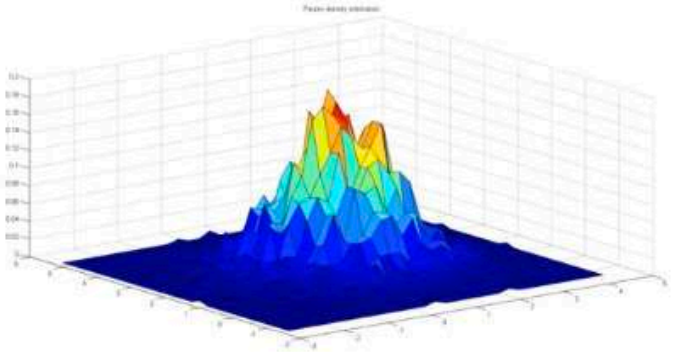
Number of modals
= 1

Mixture of Gaussian Density Estimation



1 <
Number of modals
< Number of instances

Kernel Density Estimation



Number of modals
= Number of instances

📌 In multi-dimensional data,

1. modal : 1 → 전체 데이터가 단 하나의 가우시안으로부터 생성되었다 가정
   → 이 하나의 분포의 평균벡터와 공분산행렬만 추정하면 된다.
2. 가우시안 분포를 mix한게 하나보단 많고, 전체 변수 수보다는 적게.
   보통 한자리 수로 추정
3. 각각의 객체들이 모두 각각의 가우시안 분포들의 중심임을 가정!!
   → 정상 데이터 영역의 밀도함수를 추정하겠다.

Gaussian Density Estimation

Advantages

- Intensive to scaling of the data

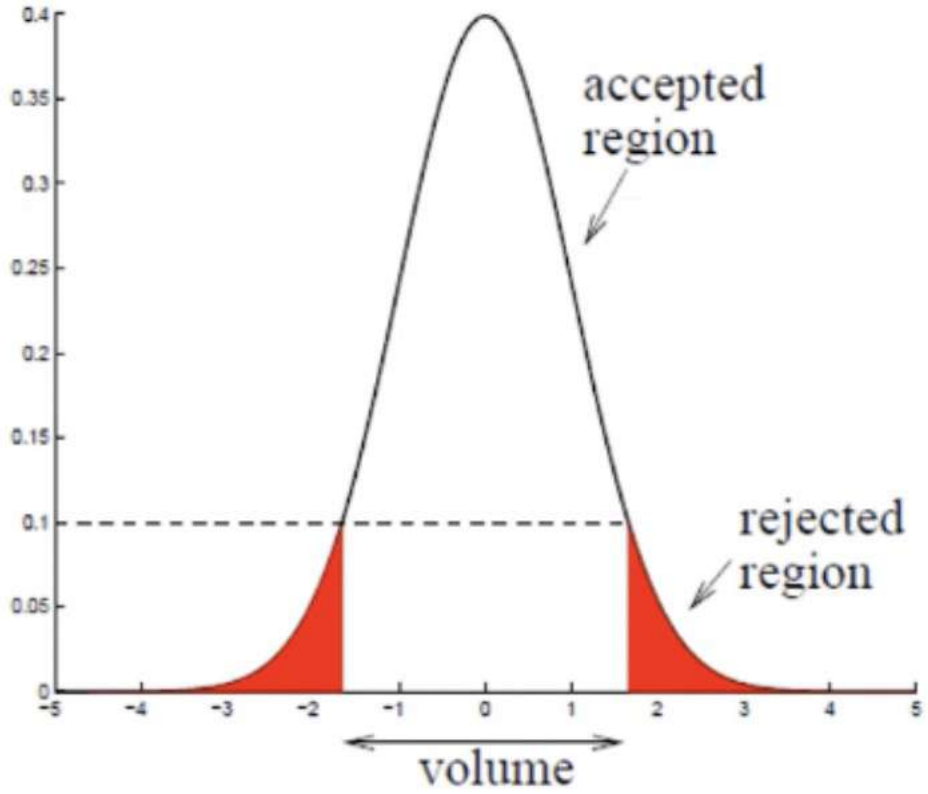
$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} exp\left[\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$

데이터의 각 변수별 분산을 고려해서 거리를 계산하기 때문에 Scaling 필요 x

Gaussian Density Estimation

Advantages

- rejection에 대한 alpha값을 정의하고 갈 수 있다.



→ possible to compute analytically the optimal threshold

cutoff를 계산할 수 있다.

Gaussian Density Estimation

Dimensions in Normal Distribution

One-dimensional

$$f_{\mu,\sigma^2}(x_i) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)$$

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2$$

Multi-Dimensional

$$p(x|\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}\exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}$$

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i, \quad \boldsymbol{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i-\boldsymbol{\mu})(\mathbf{x}_i-\boldsymbol{\mu})^T$$

-> Mean Vector & Covariance Matrix !!

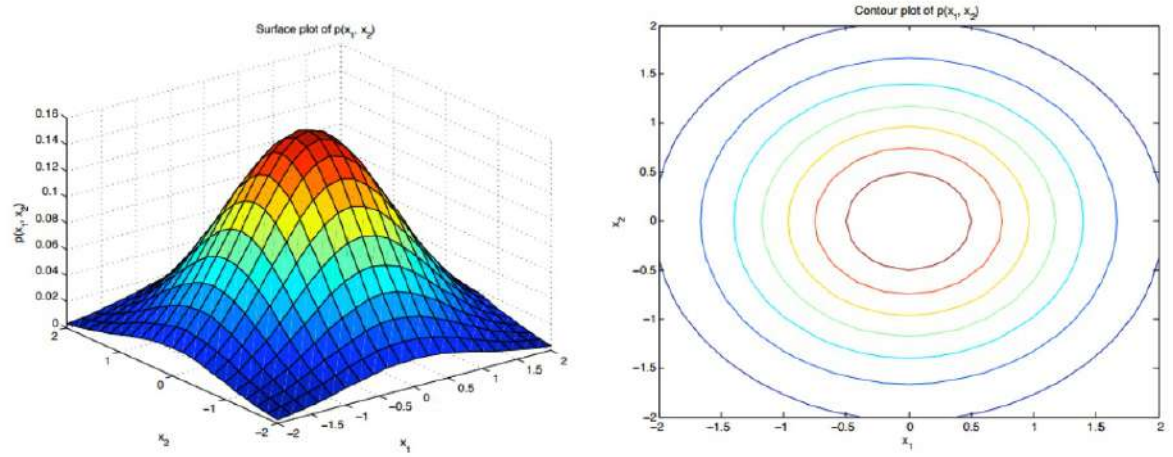
Gaussian Density Estimation

Covariance Matrix Type

Spherical Type

등고선이 원

$$\sum = \sigma^2 \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}$$

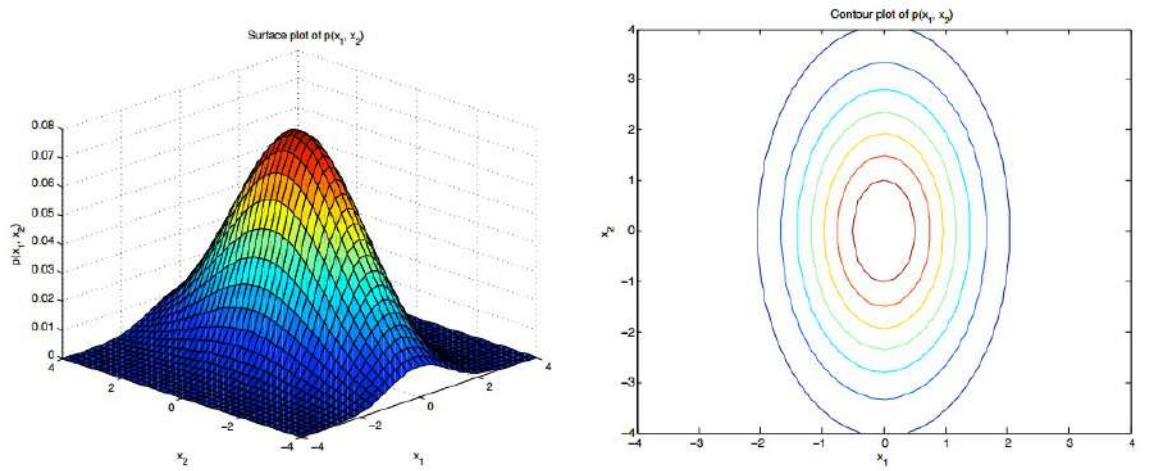


(a) Spherical Gaussian (diagonal covariance, equal variances)

Diagonal Type

변수별로 수직 유지,
변수별 값이 다르기 때문에 타원

$$\sum = \begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_d^2 \end{bmatrix}$$



(b) Gaussian with diagonal covariance matrix

Gaussian Density Estimation

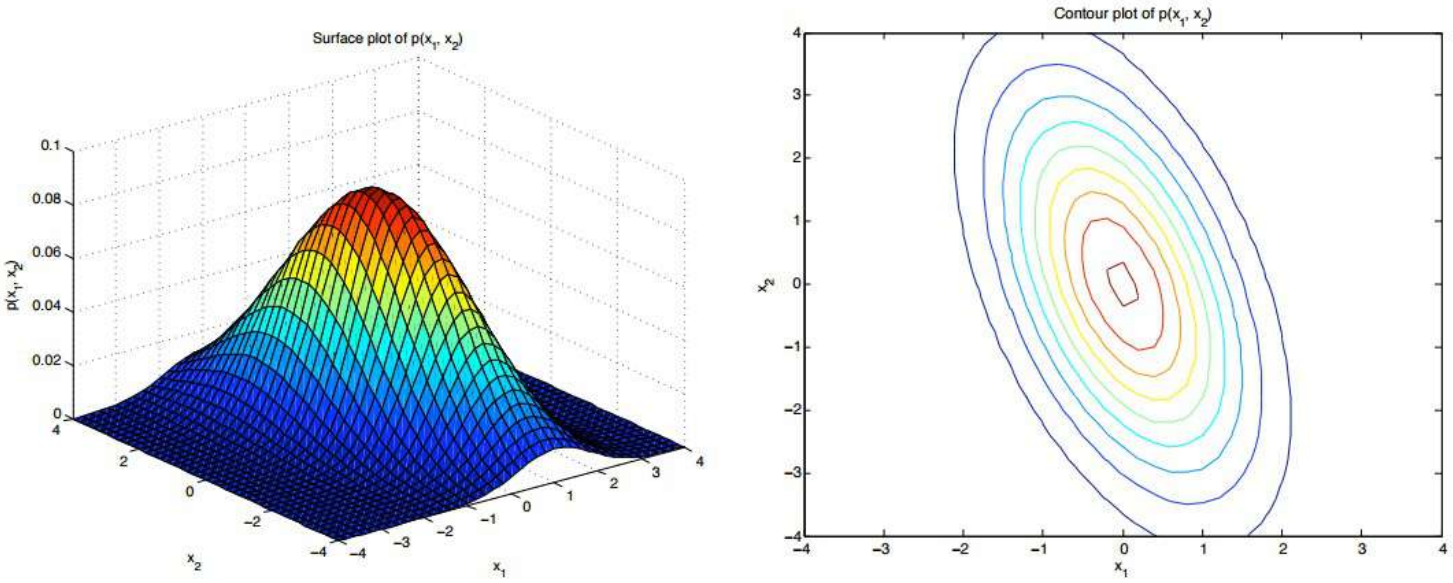
Covariance Matrix Type

Full Type

전부 고려하기 때문에 축이 기울어짐

$$\sum = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1d} \\ \vdots & \ddots & \vdots \\ \sigma_{d1} & \cdots & \sigma_{dd} \end{bmatrix}$$

변수가 너무 많아지면 공분산 행렬이 Singular Matrix(=비가역) 가능성이 높아짐 = (Det=0)



(c) Gaussian with full covariance matrix

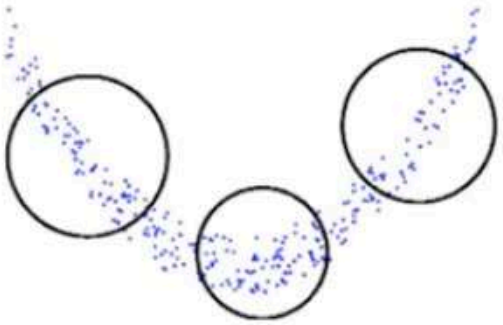
sample covariance matrix calculated when $p>n$ is singular.
rank of the covariance matrix is no greater than $\min(p,n)$

Gaussian Density Estimation

Covariance Matrix Type

Spherical

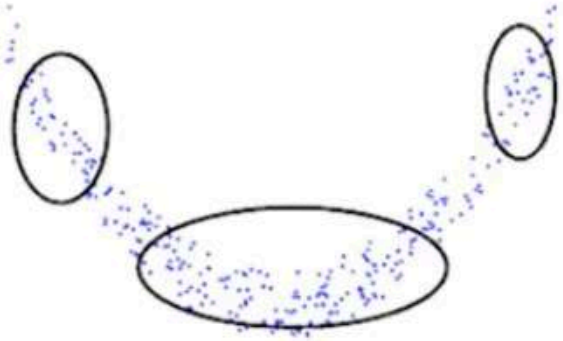
$$\Sigma = \sigma^2 \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}$$



- Less precise
- Very efficient to compute

Diagonal

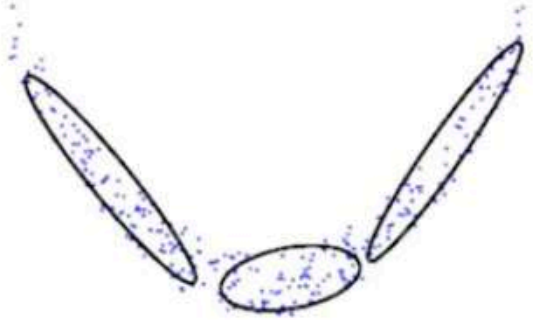
$$\Sigma = \begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_d^2 \end{bmatrix}$$



- More precise
- Efficient to compute

Full

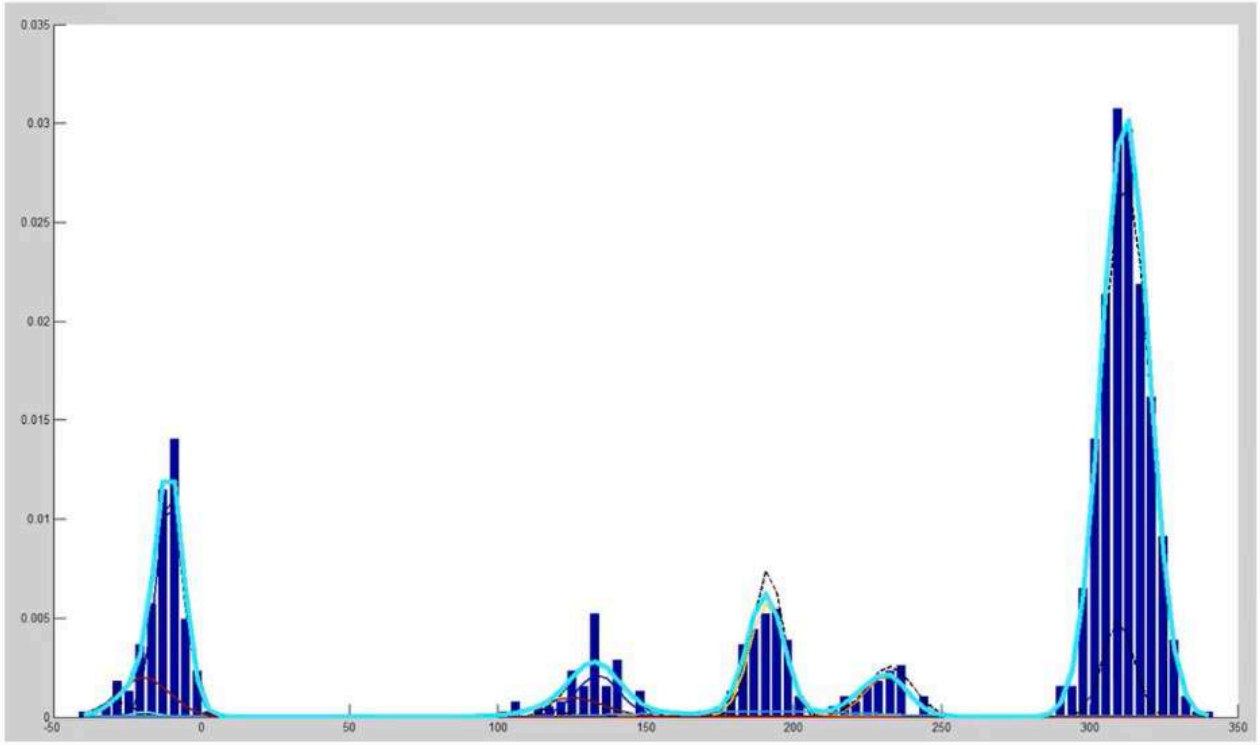
$$\Sigma = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1d} \\ \vdots & \ddots & \vdots \\ \sigma_{d1} & \cdots & \sigma_{dd} \end{bmatrix}$$



- Very precise
- Less efficient to compute

Gaussian Mixture Model

GMM?
: Gaussian Distributions' Linear Combination !



$$f(x) = w_1 N(\mu_1, \sigma_1^2) + w_2 N(\mu_2, \sigma_2^2) + w_3 N(\mu_3, \sigma_3^2) + w_4 N(\mu_4, \sigma_4^2) + w_5 N(\mu_5, \sigma_5^2)$$

Gaussian Mixture Model

Difference occuring In Mixture Model

One-dimensional

Multi-Dimensional

Single Gaussian

$$f_{\mu,\sigma^2}(x_i) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)$$

$$p(x|\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}\exp\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\}$$

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i, \quad \boldsymbol{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i-\boldsymbol{\mu})(\mathbf{x}_i-\boldsymbol{\mu})^T$$

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2$$

-> Mean Vector & Covariance Matrix !!

GMM

세 가지 모수들을 적절히 추정하는 것이 목적 <-

- Probability of an instance that belongs to the normal class

$$p(\mathbf{x}|\lambda) = \sum_{m=1}^{M} w_m g(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$$

lambda ; 미지수 집합

mixing coefficient들은 X 전체에서 X_k가 얼마나 자주 나타나는지를 확률적으로 표현

→ 각 k에 대해서 $\pi_k \geq 0, \Sigma\pi_k = 1$

Gaussian Mixture Model

$$\theta = \{\pi, \mu, \Sigma\}$$

$$\pi \equiv \{\pi_1, \ldots, \pi_K\}, \mu \equiv \{\mu_1, \ldots, \mu_K\} \; \Sigma \equiv \{\Sigma_1, \ldots \Sigma_K\}.$$

GMM can express various Density models than single Gaussian Model..

$$p(z_k = 1|\mathbf{x}_n) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\mu_j, \Sigma_j)} = \gamma(z_{nk})$$

Responsibility – 책임값..

: n번째 데이터가 어느 가우시안에서 왔을까?

$$\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Similar to Softmax Function..

Gaussian Mixture Model

Responsibility

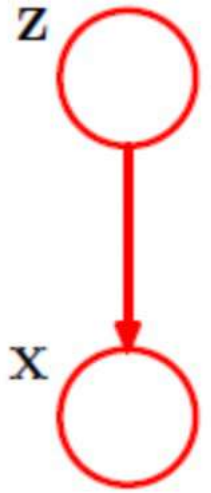
$p(z_{nk}=1|\mathbf{x}_n)$

: Xn이 K개의 Gaussian 중 어떤 가우시안에서 왔을 확률

Z : 잠재변수(Latent Variable)

; 실제 볼 수는 없지만 필요해서 가정하는 변수

x가 실제 k번째 가우시안에서 왔으면 1 그렇지 않으면 0 두가지 값



$\pi_k = p(z_k = 1)$ <- Mixing coefficient : k번째 가우시안에서 나오는 점들을 관측할 전체 확률

$\mathbf{z} = \{z_1, \ldots, z_K\}$

k개의 잠재변수 z를 모아서 벡터로 표현할 수 있다.(해당 클러스터에서만 1값을 갖는 one-hot vector)

$$p(\mathbf{z}) = p(z_1 = 1)^{z_1} p(z_2 = 1)^{z_2} \ldots p(z_K = 1)^{z_K} = \prod_{k=1}^{K} \pi_k^{z_k}$$

<- 다항분포의 분포함수!

Gaussian Mixture Model

뭘.. 구하고 있었더라..? Responsibility!

$$p(z_{nk} = 1|\mathbf{x}_n)$$

$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)} = \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$$

$$p(\mathbf{x}_n|\mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}_n|\mu_k, \boldsymbol{\Sigma}_k)^{z_k}$$

: 특정 가우시안이 정해졌을 때 해당 데이터가 나올 확률

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$\pi_k = p(z_k = 1)$ <- Mixing coefficient : k번째 가우시안에서 나오는 점들을 관측할 전체 확률

Gaussian Mixture Model

뭘 찾고 싶었더라..? Parameters!

Mean

$$\frac{\partial \mathcal{L}(X;\theta)}{\partial \mu_k} = \sum_{n=1}^{N} \frac{\pi_k N(x_n|\mu_k,\ \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j,\ \Sigma_j)} \Sigma_k^{-1}(x_n - \mu_k) = 0$$

$$0 = -\sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{\gamma(z_{nk})} \boldsymbol{\Sigma}_k(\mathbf{x}_n - \boldsymbol{\mu}_k)$$

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

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

Covariance

$$\frac{\partial \mathcal{L}(X;\theta)}{\partial \Sigma_k} = \sum_{n=1}^{N} \frac{\pi_k N(x_n|\mu_k,\ \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j,\ \Sigma_j)} \left\{ \frac{1}{2} \Sigma_k^{-1}(x_n-\mu_k)(x_n-\mu_k)^T \Sigma_k^{-1} - \frac{1}{2} \Sigma_k^{-1} \right\} = 0$$

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

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

Gaussian Mixture Model

뭘 찾고 싶었더라..? Parameters!

Mixing Coefficient by Lagrange method
; with constraints -> without constraints

$$J(X;\theta,\ \lambda) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \pi_k N(x_n|\mu_k,\ \Sigma_k) + \lambda\left(1 - \sum_{k=1}^{K} \pi_k\right)$$

$$\frac{\partial J(X;\theta,\ \lambda)}{\partial \pi_k} = \sum_{n=1}^{N} \frac{N(x_n|\mu_k,\ \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j,\ \Sigma_j)} - \lambda = 0$$

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

$$\Leftrightarrow \sum_{k=1}^{K}\sum_{n=1}^{N} \gamma(z_{nk}) - \lambda = 0 \quad \left(\because \sum_{k=1}^{K} \pi_k = 1\right)$$

$$\therefore \lambda = N \quad \left(\because \sum_{k=1}^{K} \gamma(z_{nk}) = 1\right)$$

$$\frac{\partial J(X;\theta,\ \lambda)}{\partial \pi_k} = \sum_{n=1}^{N} \frac{N(x_n|\mu_k,\ \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j,\ \Sigma_j)} - N = 0$$

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

$$\therefore \pi_k = \frac{1}{N}\sum_{n=1}^{N} \gamma(z_{nk})$$

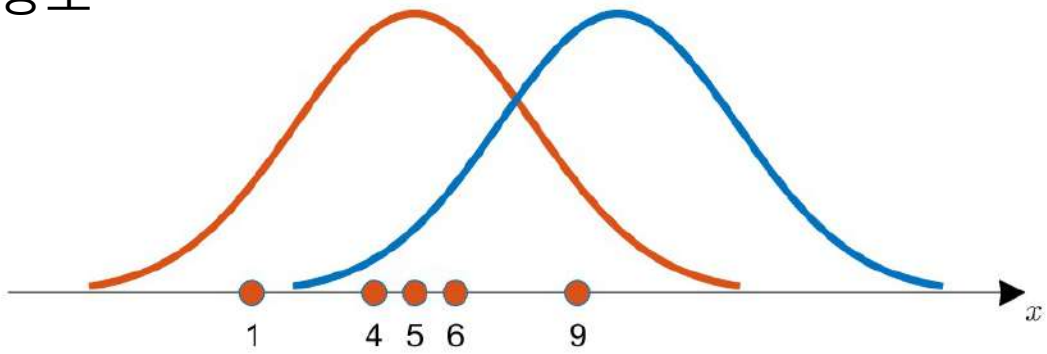
But, 이렇게 analytic하게 추정한 값들은
나머지 값들이 고정되어 있을 때만 가능.
-> EM Algorithm

Gaussian Density Estimation

EM Algorithm Introduction
Maximum Likelikelihood

Likelihood : 지금 얻은 데이터가 이 분포로부터 나왔을 가능도

<- 각 데이터 샘플에서 후보 분포에 대한 높이들의 곱



Likelihood Function

$$P(x|\theta) = \prod_{k=1}^{n} P(x_k|\theta) \qquad L(\theta|x) = \log P(x|\theta) = \sum_{i=1}^{n} \log P(x_i|\theta)$$

$$\frac{\partial}{\partial\theta} L(\theta|x) = \frac{\partial}{\partial\theta} \log P(x|\theta) = \sum_{i=1}^{n} \frac{\partial}{\partial\theta} \log P(x_i|\theta) = 0$$

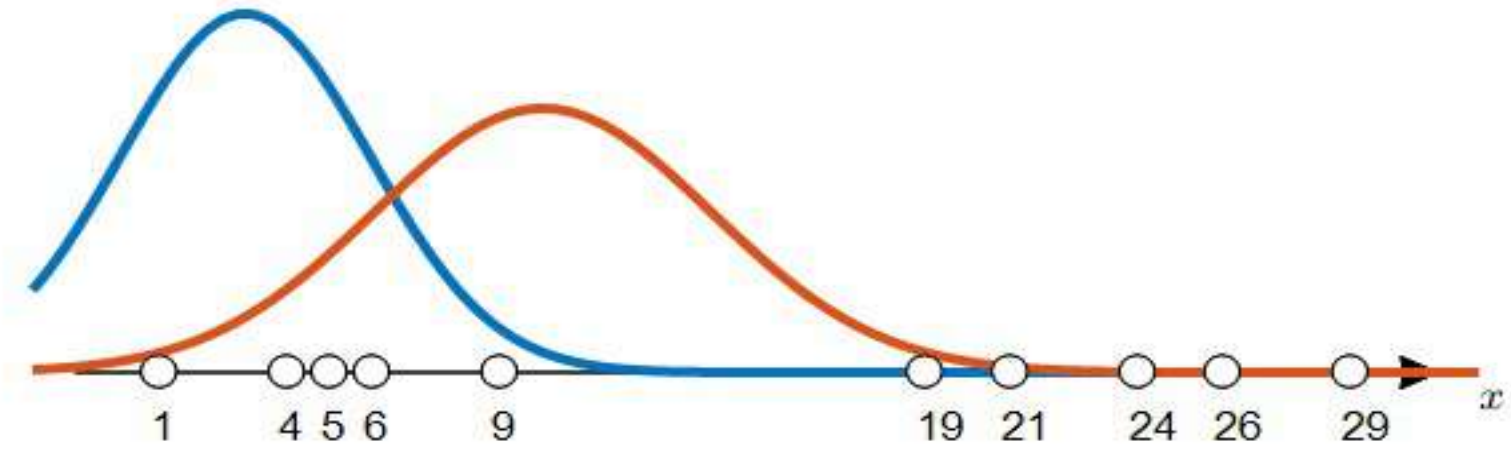
Gaussian Mixture Model

EM Algorithm Introduction

확률 분포를 얻기 위해선 모수를 알아야 하고, 모수를 알기 위해서는 label을 알아야 한다.

Gaussian Mixture Model

EM Algorithm Introduction



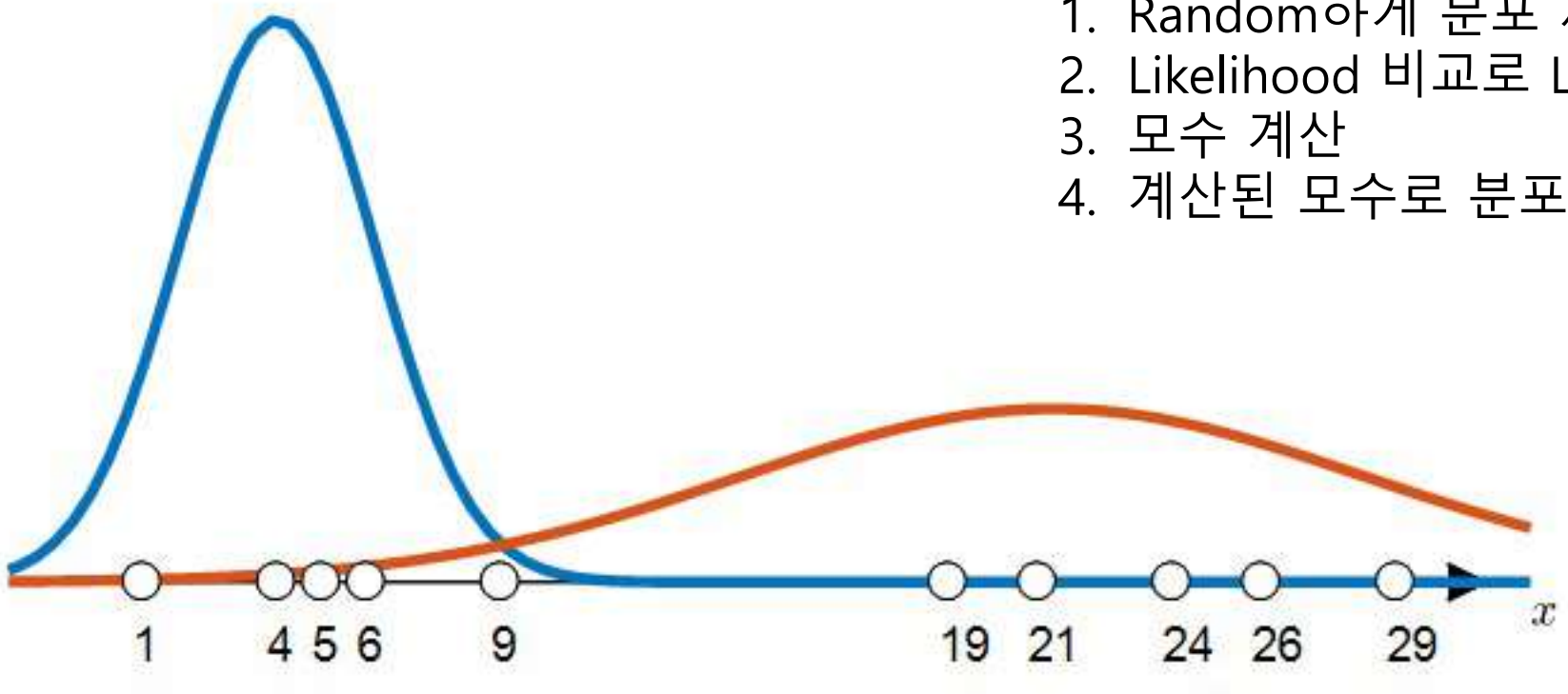
우선 random하게 분포를 그려보자. 그러면 label 생성은 가능할 것

$$\mu_1 = 4, \sigma_1 = 2.1602$$

$$\mu_2 = 21.33, \sigma_2 = 7.0048$$

Gaussian Mixture Model

EM Algorithm Introduction



1. Random하게 분포 제안
2. Likelihood 비교로 Label 부여
3. 모수 계산
4. 계산된 모수로 분포 다시

Gaussian Mixture Model

EM Algorithm Introduction

w_m, μ_m, Σ_m 을 찾고 싶다.

- E-Step

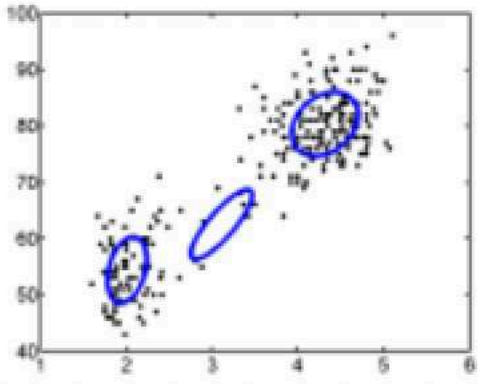
Gaussian을 고정한 상태에서 각각의 객체들이 각 Gaussian에 속할 확률을 추정.

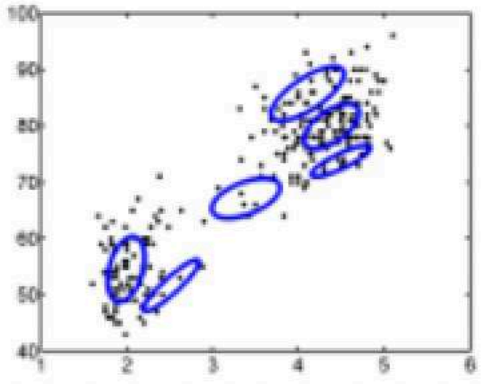
- M-Step

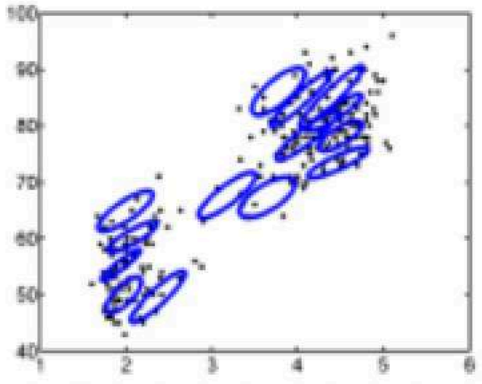
객체의 확률을 고정한 다음에 Gaussian의 parameter를 업데이트

↔️ 반복하다 보면... 최적의 Gaussian분포의 세 미지수를 찾을 수 있을거야!

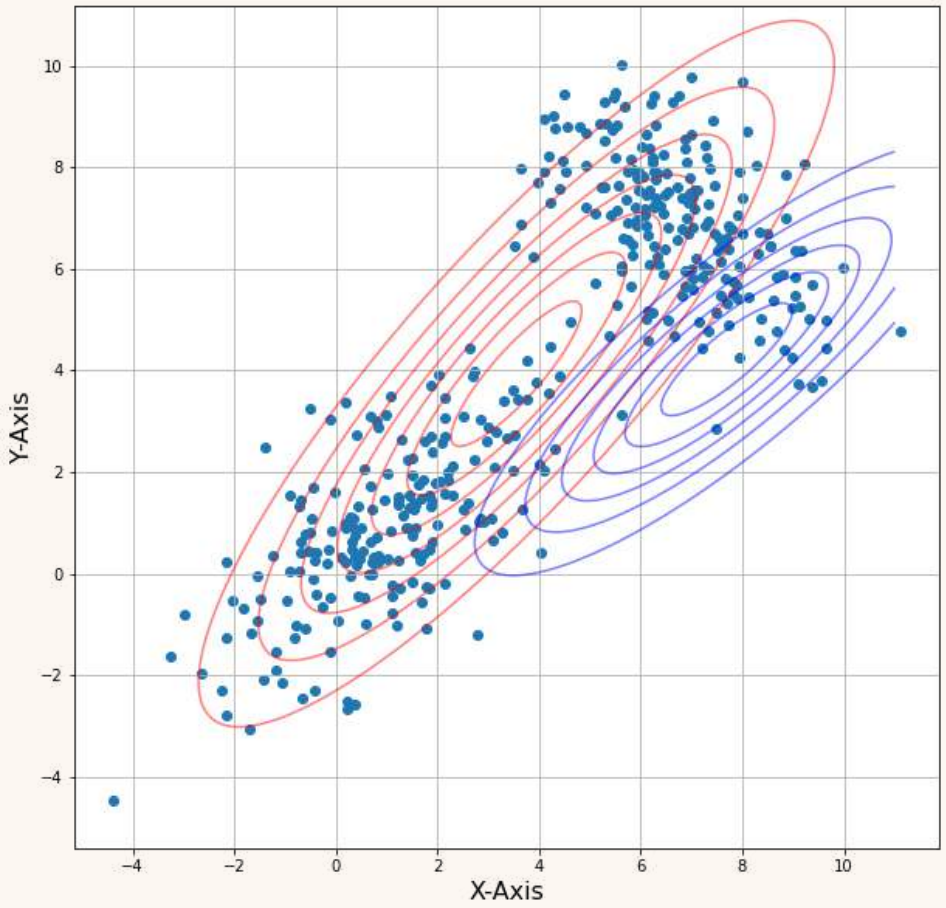
1. Random하게 분포 제안
2. Likelihood 비교로 Label 부여
3. 모수 계산
4. 계산된 모수로 분포 다시

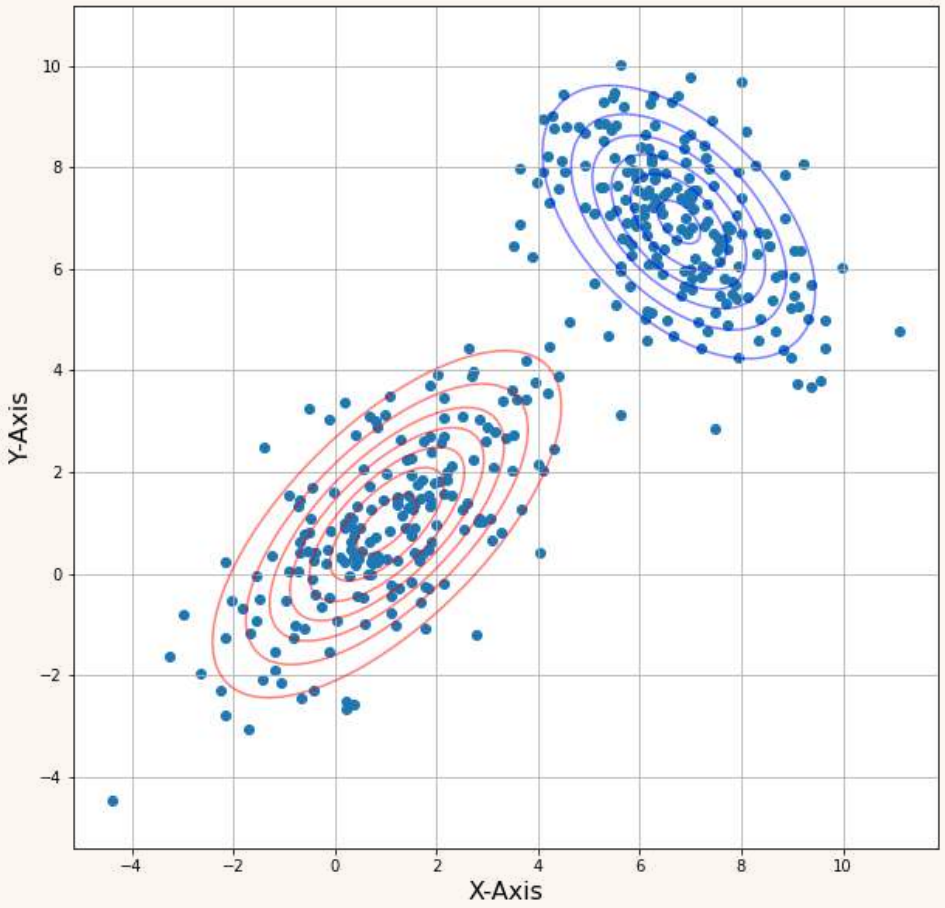






Homework – Coding





Contents

Probabilistic Machine Learning

- From observed variables & unobserved variables, we want to know how the known variables are related to the unknown variables by certain **models**.

- Probabilistic model: joint distribution of hidden variables z and observed variables x

- And we want to know posterior from the model

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

Probabilistic Machine Learning

- But… denominator $p(x)$ is often too complicated
- So, we approximate posterior

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

4. 현재 메시지함에 5통의 메시지가 있습니다. 5통의 메시지를 바탕으로 앞으로의 메시지가 스팸인지를 판별해주는 베이지안 분류모델을 구축하려고 합니다. 이 때 특정 문자에 '무료'라는 단어가 있을 때 해당 문자가 스팸문자일 확률은 얼마입니까? ham은 일반문자이며 spam은 스팸문자 입니다.

spam : 대출 전화 안내
spam : 무료 대출 지금 전화
spam : 지금 무료 지급
ham : 장학금 무료 지급 안내
ham : 요금 납부 안내

Probabilistic Inference

• Compute a desired probability from other known probability

Ex)

S	T	W	P
summer	hot	sun	0.30
summer	hot	rain	0.05
summer	cold	sun	0.10
summer	cold	rain	0.05
winter	hot	sun	0.10
winter	hot	rain	0.05
winter	cold	sun	0.15
winter	cold	rain	0.20

- $P(W)$
- $P(W \mid \text{winter})$
- $P(W \mid \text{winter, hot})$

Probabilistic Inference

In general, given n variables $X_1, \ldots X_n$, we want to obtain $P(Q|e_1, \ldots e_k)$

where $(E_1, \ldots E_k) = (e_1, \ldots e_k)$ are evidence

and $H_1, \ldots H_r$ are hidden variables.

$$P(Q|e_1, \ldots e_k) = \frac{1}{Z} P(Q, e_1, \ldots e_k)$$

where $Z = \sum_q P(Q, e_1, \ldots e_k)$

and $P(Q, e_1, \ldots e_k) = \sum_{h_1, \ldots h_r} P(Q, h_1, \ldots h_r, e_1, \ldots e_k)$

Probabilistic Inference

But, it would be also hard to compute…

So we use such ways.

- Reduce joint probability table by assuming (conditional) independence, causality, Markov property, **latent variables**, etc.
- Approximate posterior inference such as **variational inference**, Monte Carlo sampling, etc.

Variational Inference

- "variational": from 'calculus of variations' by Euler, Lagrange, ..
  : using functionals
  : refers to tools for optimization-based formulations of problem

- Idea : express a quantity of interest as the solution of optimization

Variational Inference

- Goal : approximate posterior by optimization!

1. Let Q be a family of approximate densities (over latent variables)
2. Find a density q such that minimizes **Kullback Leibler divergence** to the exact posterior

$$q(z) = argmin D_{KL}[q(z)||p(z|x)]$$

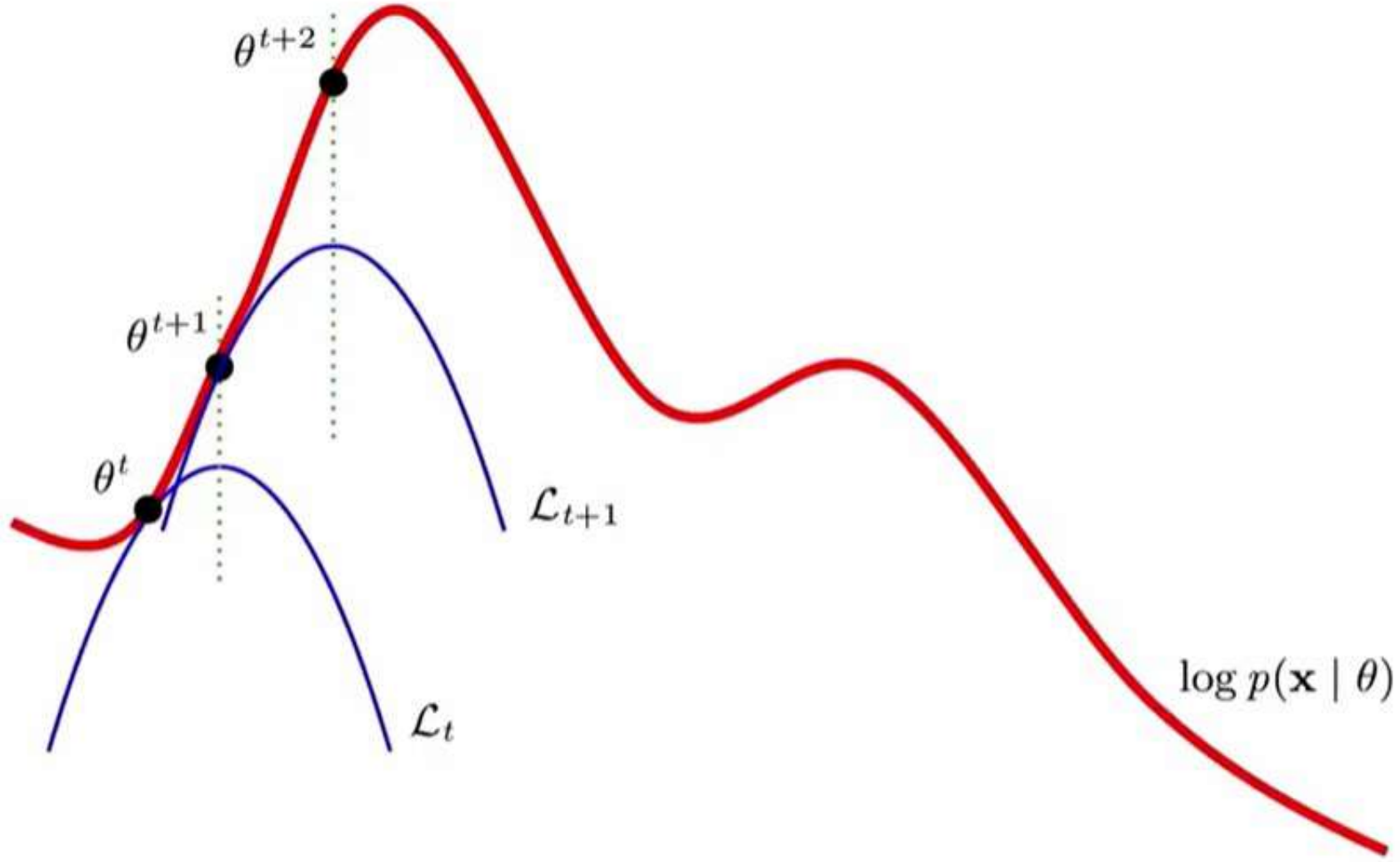
where $q \in Q$, and $D_{KL}[Q||P)] = E_{x\sim Q}[\log Q - \log P]$

3. Approximate the posterior with q

EM Algorithm

- Goal : find θ such that maximizes likelihood $p(X|\theta)$ or $\log p(X|\theta)$
- But such likelihood is too complicated to optimize… (no closed form, multiple local maxima, …)
- So we use EM algorithm (Expectation - Maximization)

EM Algorithm



EM Algorithm

1. Guess the initial θ

Repeat 2~3 until convergence

2. (E-step) Choose function $\mathcal{L}_t$ (called ELBO)which is lower bound of $\log p(x|\theta^t)$

So, $\mathcal{L}_t(\theta^t) = \log p(x|\theta^t)$

3. (M-step) Change the parameter θ^t to θ^{t+1} such that maximizes $\mathcal{L}_t$

So, $\log p(x|\theta^t) = \mathcal{L}_t(\theta^t) \leq \mathcal{L}_t(\theta^{t+1}) = \log p(x|\theta^{t+1})$

Latent Variable Model

- Fewer parameters, latent variables work as bottleneck
- But, hard to fit …

- We use such model

- when there are hidden observations or missing value (incomplete data)
- when it is hard to optimize the posterior directly

EM Algorithm

- Goal : find θ such that maximizes likelihood $p(X|\theta)$ or $\log p(X|\theta)$

- We use latent variable : $p(X|\theta) = \sum_Z p(X, Z|\theta)$

for EM algorithm (Expectation - Maximization)

Remark) EM obtains **local** maximum only, so we need to run EM using **multiple initial** parameters with different values.

Example for simple EM algorithm

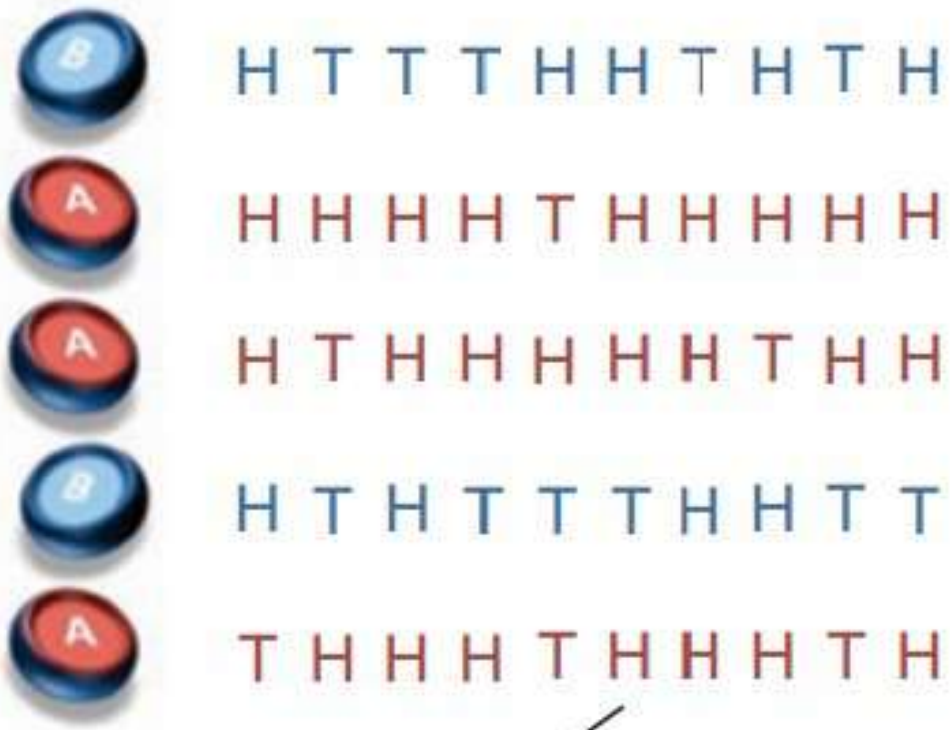
Given coins A and B with prob. θ_A and θ_B of landing on heads.

Our goal is to estimate $\theta = (\theta_A, \theta_B)$

(1) randomly choose one of the two coins

(2) perform 10 independent coin tosses with the selected coin

Case 1) complete data

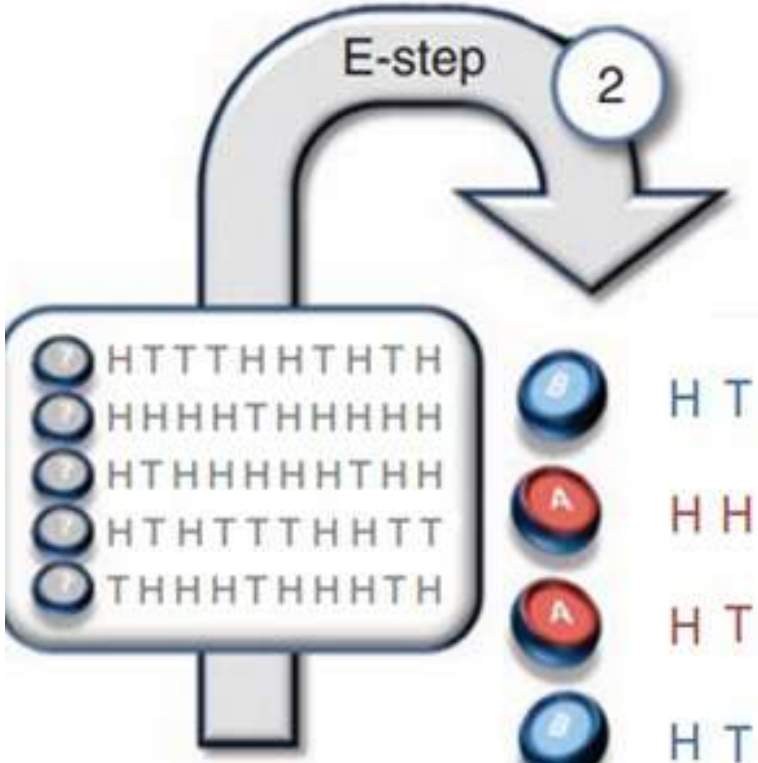


Coin A	Coin B
	5 H, 5 T
9 H, 1 T	
8 H, 2 T	
	4 H, 6 T
7 H, 3 T	
24 H, 6 T	9 H, 11 T

$$\hat{\theta}_A = \frac{24}{24+6} = 0.80$$

$$\hat{\theta}_B = \frac{9}{9+11} = 0.45$$

Case 2) incomplete data – hard version



Coin A	Coin B
	5 H, 5 T
9 H, 1 T	
8 H, 2 T	
	4 H, 6 T
7 H, 3 T	
24 H, 6 T	9 H, 11 T

H T T T H H T H T H

H H H H T H H H H H

H T H H H H H T H H

H T H T T T H H T T

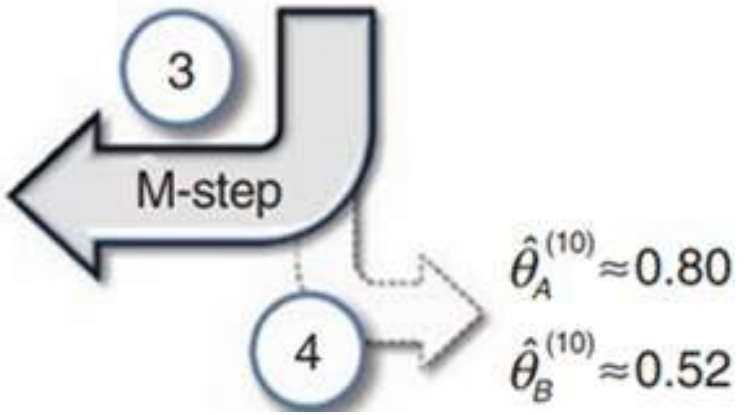
T H H H T H H H T H

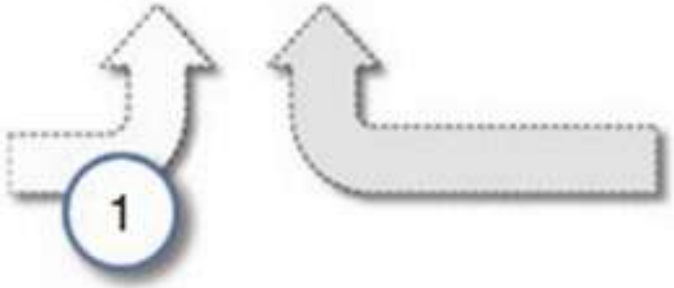
$\hat{\theta}_A^{(0)} = 0.60$

$\hat{\theta}_B^{(0)} = 0.50$

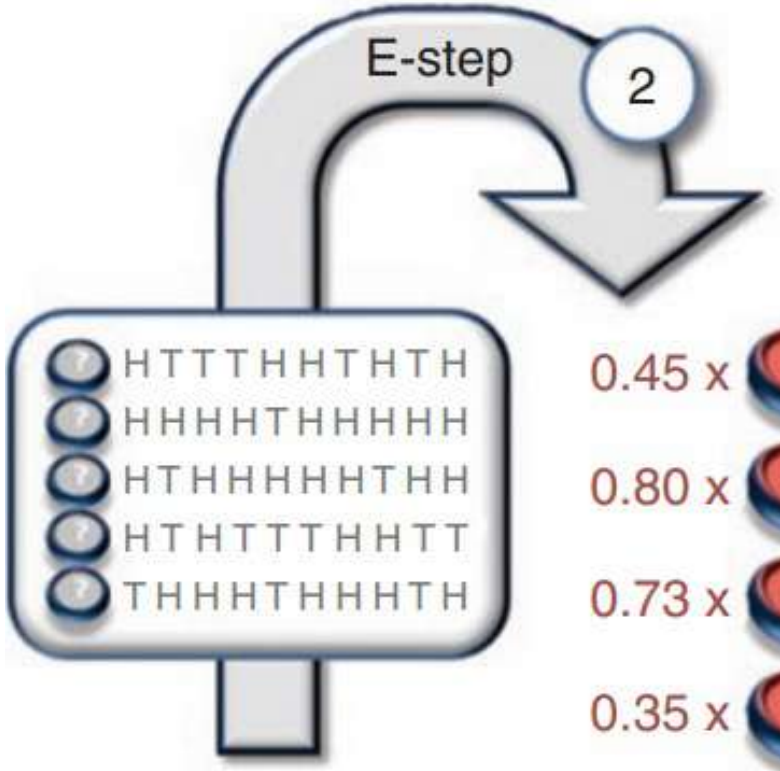
$$\hat{\theta}_A = \frac{24}{24+6} = 0.80$$

$$\hat{\theta}_B = \frac{9}{9+11} = 0.45$$





Case 3) incomplete data – soft version



	Coin A	Coin B
0.45 x A + 0.55 x B	≈ 2.2 H, 2.2 T	≈ 2.8 H, 2.8 T
0.80 x A + 0.20 x B	≈ 7.2 H, 0.8 T	≈ 1.8 H, 0.2 T
0.73 x A + 0.27 x B	≈ 5.9 H, 1.5 T	≈ 2.1 H, 0.5 T
0.35 x A + 0.65 x B	≈ 1.4 H, 2.1 T	≈ 2.6 H, 3.9 T
0.65 x A + 0.35 x B	≈ 4.5 H, 1.9 T	≈ 2.5 H, 1.1 T
	≈ 21.3 H, 8.6 T	≈ 11.7 H, 8.4 T

$\hat{\theta}_A^{(0)} = 0.60$

$\hat{\theta}_B^{(0)} = 0.50$

1

$$\hat{\theta}_A^{(1)} \approx \frac{21.3}{21.3 + 8.6} \approx 0.71$$

$$\hat{\theta}_B^{(1)} \approx \frac{11.7}{11.7 + 8.4} \approx 0.58$$

3

M-step

4

$\hat{\theta}_A^{(10)} \approx 0.80$

$\hat{\theta}_B^{(10)} \approx 0.52$

GMM with EM Algorithm

- $Z \in \{1, \ldots, K\}$: discrete latent dwhose prior is $P(Z = k) = \pi_k$
- $p(x|Z = k) = p_k(x) = N(x|\mu_k, \Sigma_k)$: likelihood

  Then, $p(x) = \sum_1^K P(Z = k)p(x|Z = k) = \sum_1^K \pi_k p_k(x)$
- $\gamma_k = P(Z = k|x)$: responsibility, posterior

  $\gamma_k(x) = P(Z = k|x) = \pi_k N(x|\mu_k, \Sigma_k) / \sum_i \pi_i N(x|\mu_i, \Sigma_i)$

Algorithm 8.1 *EM Algorithm for Two-component Gaussian Mixture.*

1. Take initial guesses for the parameters $\hat{\mu}_1, \hat{\sigma}_1^2, \hat{\mu}_2, \hat{\sigma}_2^2, \hat{\pi}$ (see text).

2. *Expectation Step*: compute the responsibilities

$$\hat{\gamma}_i = \frac{\hat{\pi}\phi_{\hat{\theta}_2}(y_i)}{(1-\hat{\pi})\phi_{\hat{\theta}_1}(y_i) + \hat{\pi}\phi_{\hat{\theta}_2}(y_i)}, \; i = 1, 2, \ldots, N. \tag{8.42}$$

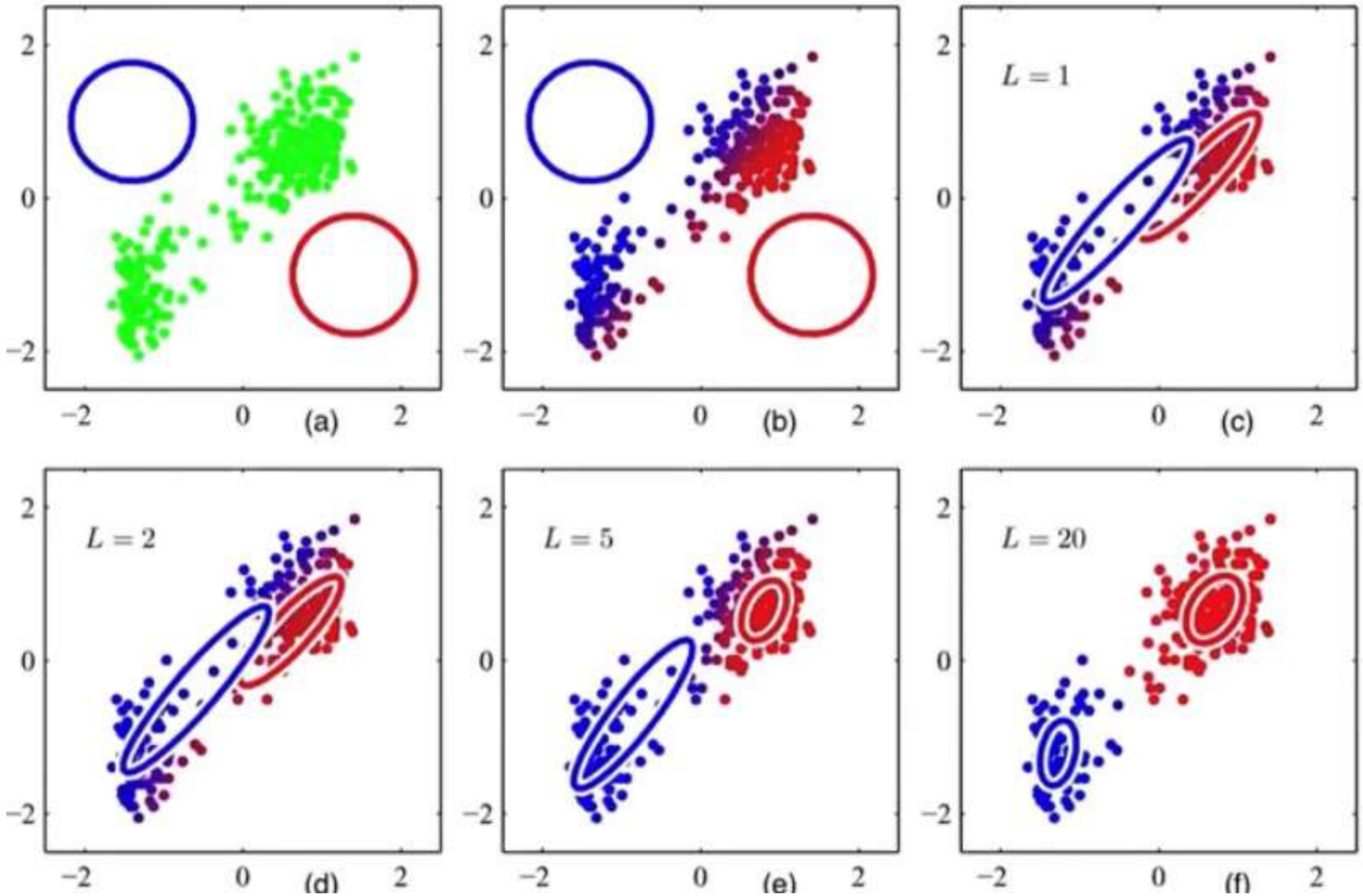
3. *Maximization Step*: compute the weighted means and variances:

$$\hat{\mu}_1 = \frac{\sum_{i=1}^N (1-\hat{\gamma}_i) y_i}{\sum_{i=1}^N (1-\hat{\gamma}_i)}, \qquad \hat{\sigma}_1^2 = \frac{\sum_{i=1}^N (1-\hat{\gamma}_i)(y_i - \hat{\mu}_1)^2}{\sum_{i=1}^N (1-\hat{\gamma}_i)},$$

$$\hat{\mu}_2 = \frac{\sum_{i=1}^N \hat{\gamma}_i y_i}{\sum_{i=1}^N \hat{\gamma}_i}, \qquad \hat{\sigma}_2^2 = \frac{\sum_{i=1}^N \hat{\gamma}_i (y_i - \hat{\mu}_2)^2}{\sum_{i=1}^N \hat{\gamma}_i},$$

   and the mixing probability $\hat{\pi} = \sum_{i=1}^N \hat{\gamma}_i / N$.

4. Iterate steps 2 and 3 until convergence.


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

EM algorithm

Our goal: maximize the likelihood function $p(X|\theta) = \sum_Z p(X, Z|\theta)$

Then $\log p(X|\theta) = \log \{\sum p(X, Z|\theta)\}$

Use Jensen's inequality for

$$\log p(X|\theta) = \log \{\sum p(X, Z|\theta)\} = \log \sum q(Z) \frac{P(X,Z|\theta)}{q(Z)}$$

EM algorithm

Then

$$\begin{aligned}\log \sum q(Z) \frac{P(X,Z|\theta)}{q(Z)} &\geq \sum q(Z) \log \frac{P(X,Z|\theta)}{q(Z)} \\ &= \sum q(Z) \log P(X,Z|\theta) - q(Z) \log q(Z) \\ &= E_{q(Z)} \log P(X,Z|\theta) + H(q)\end{aligned}$$

Here, $H(q) = -\sum q(Z) \log q(Z)$ is entropy.

Let $\mathcal{L}(\theta, q) = E_{q(Z)} \log P(X,Z|\theta) + H(q)$

EM algorithm

Then

$$\begin{aligned}\log \sum q(Z) \frac{P(X,Z|\theta)}{q(Z)} &\geq \sum q(Z) \log \frac{P(X,Z|\theta)}{q(Z)} \\ &= \sum q(Z) \log P(X,Z|\theta) - q(Z) \log q(Z) \\ &= E_{q(Z)} \log P(X,Z|\theta) + H(q)\end{aligned}$$

Here, $H(q) = -\sum q(Z) \log q(Z)$ is entropy.

Let $\mathcal{L}(\theta, q) = E_{q(Z)} \log P(X,Z|\theta) + H(q)$

EM algorithm

Then

$$\begin{aligned}
\mathcal{L}(\theta, q) &= \sum q(Z) \log \frac{P(X,Z|\theta)}{q(Z)} \\
&= \sum q(Z) \log \frac{P(Z|X,\theta)P(X|\theta)}{q(Z)} \\
&= \sum \left(q(Z) \log \frac{P(Z|X,\theta)}{q(Z)} + q(Z) \log P(X|\theta) \right) \\
&= \sum q(Z) \log \frac{P(Z|X,\theta)}{q(Z)} + \log P(X|\theta) \\
&= \log P(X|\theta) - \sum q(Z) \log \frac{q(Z)}{P(Z|X,\theta)}
\end{aligned}$$

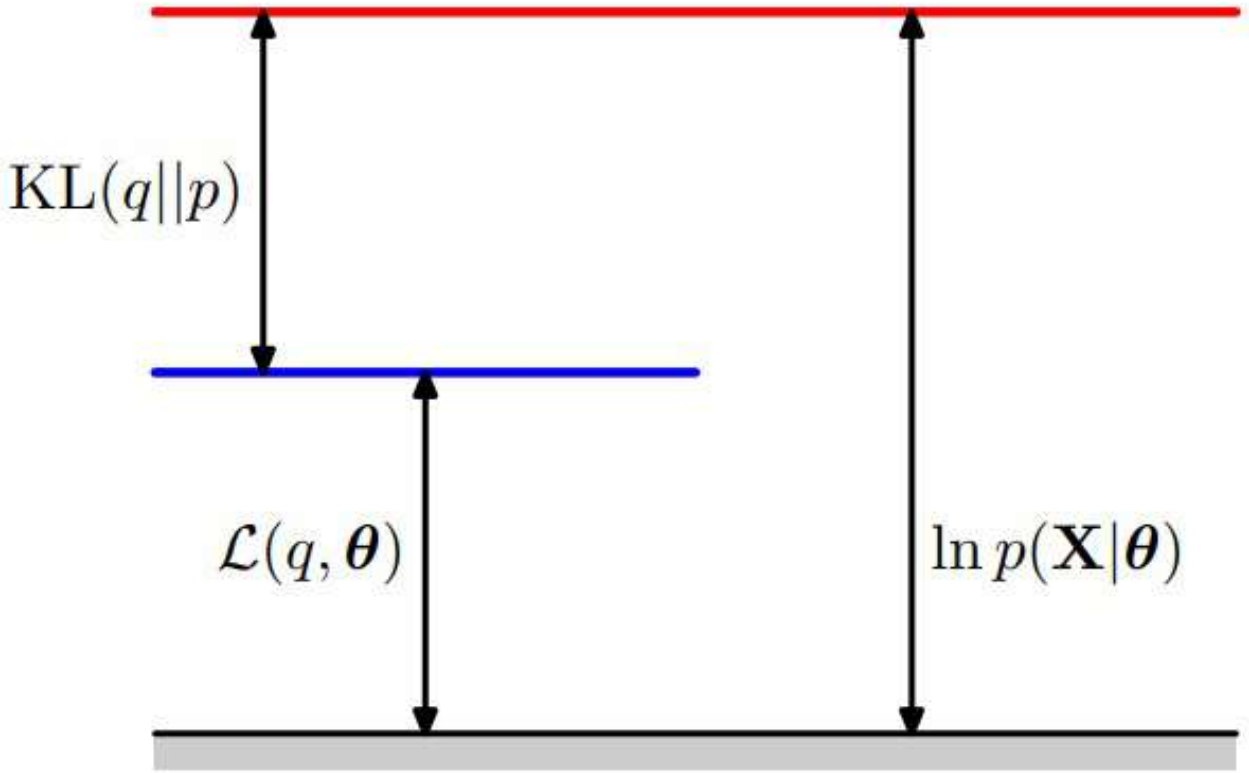
Here,

$$-\sum_Z q(Z) \log \{p(Z|X,\theta)/q(Z)\} = KL(q||p)$$

EM algorithm

So, we finally have

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



EM algorithm

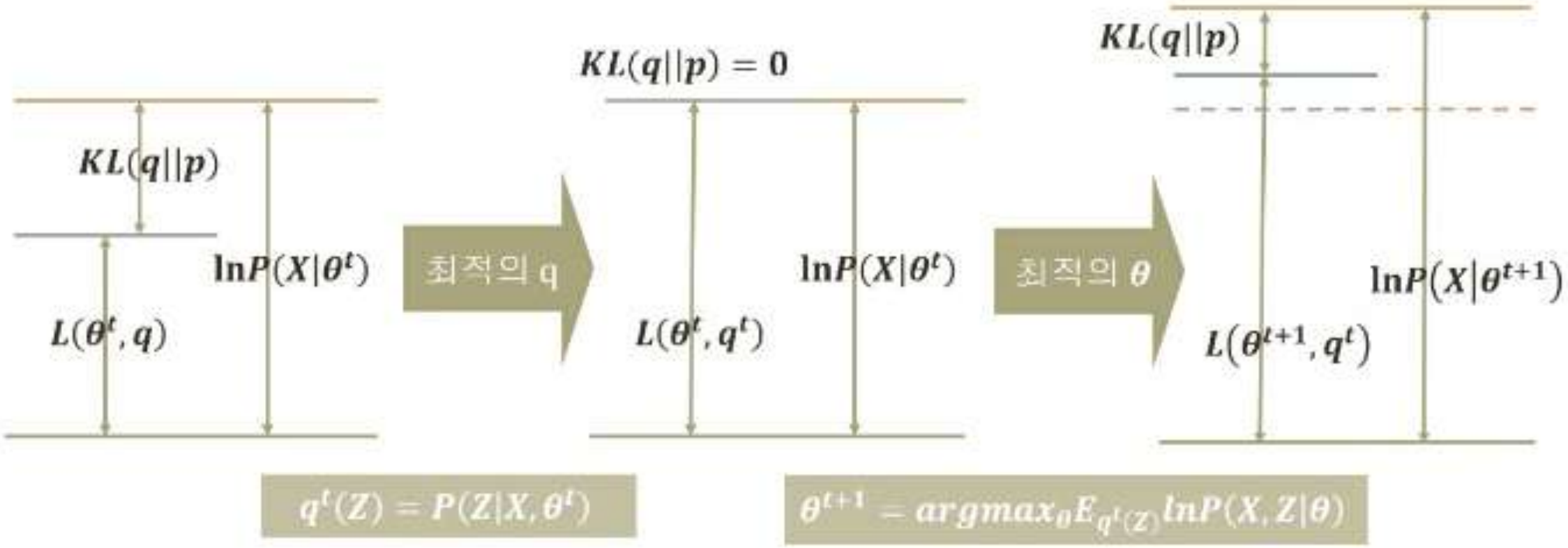


그림 8-15: EM 알고리즘의 반복

Homework - ESL

Ex. 8.1 Let $r(y)$ and $q(y)$ be probability density functions. Jensen's inequality states that for a random variable X and a convex function $\phi(x)$, $\mathrm{E}[\phi(X)] \geq \phi[\mathrm{E}(X)]$. Use Jensen's inequality to show that

$$\mathrm{E}_q \log[r(Y)/q(Y)] \tag{8.61}$$

is maximized as a function of $r(y)$ when $r(y) = q(y)$. Hence show that $R(\theta, \theta) \geq R(\theta', \theta)$ as stated below equation (8.46).

References

- The Elements of Statistical Learning (Hastie, Tibshirani) Ch.8
- Pattern Recognition And Machine Learning (Bishop) Ch.1, 9
- "What is the expectation maximization algorithm?(Chuong B Do & Serafim Batzoglou, Nature Biotechnology)
- https://sanghyu.tistory.com/16
- https://angeloyeo.github.io/2021/02/08/GMM_and_EM.html
- https://github.com/aailabkaist/Introduction-to-Artificial-Intelligence-Machine-Learning
- https://ratsgo.github.io/generative%20model/2017/12/19/vi/

Q&A, comments