EM-Algorithm &
EM Algorithm with Missing Data and its extensions

ESC 2024 Summer Session 5차
발제자: 권도현 김소민 유원희 정예준

Contents

1

EM Algorithm

1.1 MLE and EM Algorithm

- **Likelihood** : 관찰된 데이터 $X = (x_1, \dots, x_n)$ 가 주어졌을 때, 데이터가 어떤 분포를 따르는 지에 대한 측도

$$p(X|\theta) = \prod_{n=1}^{N} p(x_n|\theta)$$

- **Log-likelihood** : the natural logarithm of likelihood function

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

- **MLE(Maximum likelihood estimator)** : likelihood 또는 log-likelihood를 최대화하는 θ 값

$$\theta_{MLE} = \arg\max_{\theta} \log P(X|\theta) = \arg\max_{\theta} \log \prod_i P(x_i|\theta) = \arg\max_{\theta} \sum_i \log P(x_i|\theta)$$

⇒ 일반적으로 **MLE** 구하는 법 **: likelihood** 또는 **log-likelihood**를 구하고, 미분하여 **0**이 되는 θ 값을 찾는다.

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

MLE를 구하고 싶은데, $L(\theta|X)$를 직접적으로 최대화할 수 없을 때는 어떻게 해야 할까? ⇒ 해결법 **: EM-Algorithm!**

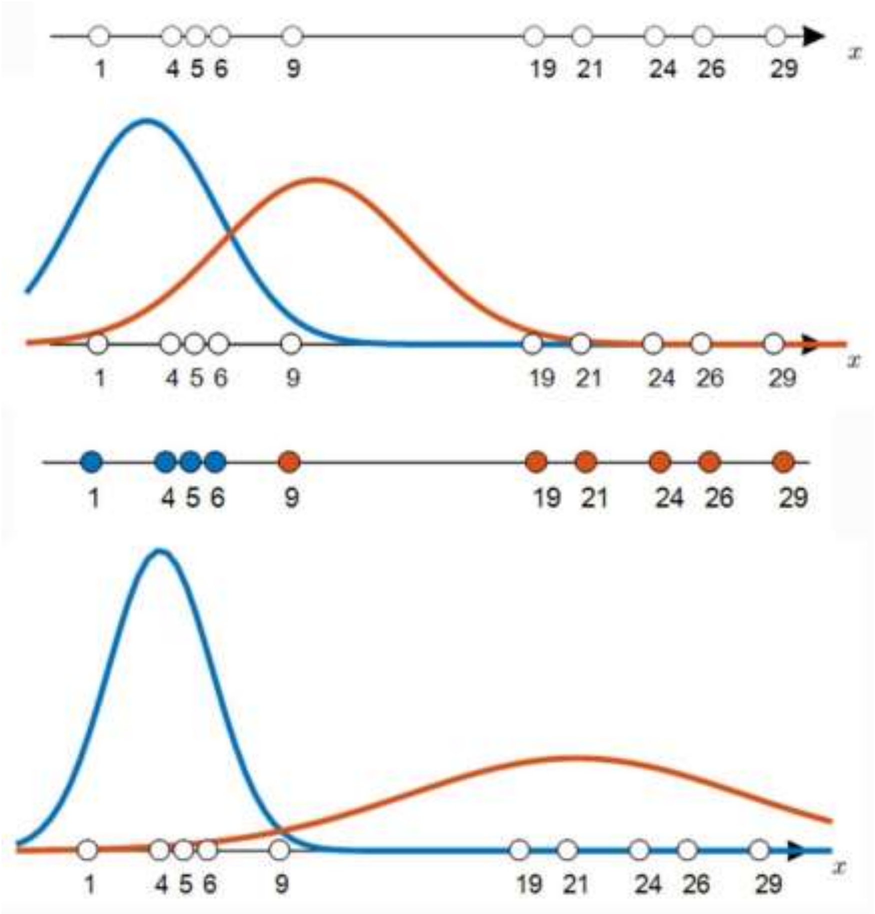
1.1 MLE and EM Algorithm

- **EM Algorithm :** 데이터 X가 주어졌을 때, 잠재변수 Z 를 통해 MLE를 구하는 Iterative 알고리즘
  파라미터가 수렴할 때까지 **E-step**과 **M-step**을 반복적으로 적용하여 **MLE**에 가까워지는 방법

✓ **Latent Variable** Z : discrete / countiuous variable 모두 가능
왜 **Latent Variable** Z를 도입할까?
$p(X|\theta)$를 직접 구하기 어렵고, $p(X,Z|\theta)$는 쉽게 계산 가능할 때,
$p(X|\theta) = \sum_Z p(X,Z|\theta)$의 식으로 $p(X|\theta)$에 접근하기 위해서!

E-step : parameter를 고정하고 latent variable를 통계적으로 할당하는 과정
M-step : 할당한 결과를 바탕으로 parameter를 다시 산정하는 과정

<예시- 분류문제>
목표 : 데이터 **10**개에 대한 **2**개의 정규분포 추정 $N_1(\mu_1, \sigma_1^2)$, $N_2(\mu_{2,}\sigma_2^2)$
Parameter : $\theta = (\mu_1, \mu_2, \sigma_1, \sigma_2)$
Latent Variable : Z=0은 N_1, **Z=1**은 N_2



1.2 EM Algorithm using B function

$L(\theta|X)$를 계산하기 어려울 때, $L(\theta|X)$를 최대화하는 θ를 어떻게 구할까?

⇒ **IDEA :** θ를 고정한 상태에서 $L(\theta|X)$와 비슷한 함수를 만들고, 그 함수를 최대화하는 값으로 θ를 다시 선정하는 과정을 θ가 수렴할 때까지 반복한다.

- **Notation**

$X = (x_1, \dots, x_n)$: 관찰데이터

$L(\theta|X)$: loglikelihood function

$q(Z)$: latent variable Z의 확률 분포

- **KL-divergence**

$KL(q \parallel p)$: 확률분포 p와 q가 얼마나 다른지를 나타내는 지표

두 분포가 동일하다면, $KL(q \parallel p) = 0$

두 분포가 동일하지 않다면, $KL(q \parallel p) > 0$

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x\in\mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right),$$

which is equivalent to

$$D_{\mathrm{KL}}(P \parallel Q) = -\sum_{x\in\mathcal{X}} P(x) \log\left(\frac{Q(x)}{P(x)}\right).$$

$$L(\theta|X) = ln\ p(X|\theta) = B(q,\theta) + KL(q \parallel p)$$

$B(q,\theta) = \sum_Z q(Z) ln \frac{P(X,Z|\theta)}{q(Z)}$: Evidence lower bound (ELBO)

$KL(q \parallel p) = -\sum_Z q(Z) ln \frac{p(Z|X,\theta)}{q(Z)}$: Kullback-Leibler divergence (KL-divergence)

1.2 EM Algorithm using B function

(1) $L(\theta|X) = B(q,\theta) + KL(q \parallel p)$

Proof

$$
\begin{aligned}
L(\theta|X) - B(q,\theta) &= \ln p(\mathbf{X}|\theta) - \textstyle\sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X},\mathbf{Z}|\theta)}{q(\mathbf{Z})} \right\} \\
&= \ln p(\mathbf{X}|\theta) - \textstyle\sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X},\theta)p(\mathbf{X}|\theta)}{q(\mathbf{Z})} \right\} \\
&= \ln p(\mathbf{X}|\theta) - \textstyle\sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X},\theta)}{q(\mathbf{Z})} \right\} - \ln p(\mathbf{X}|\theta) \textstyle\sum_{\mathbf{Z}} q(\mathbf{Z}) \\
&= - \textstyle\sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X},\theta)}{q(\mathbf{Z})} \right\} \\
&= KL[q(\mathbf{Z}) \| p(\mathbf{Z}|\mathbf{X},\theta)] \\
&= KL[q\|p]
\end{aligned}
$$

(2) $L(\theta|X) \geq B(q,\theta)$

$\Leftrightarrow$ **Lower bound of** $L(\theta|X)$ **is** $B(q,\theta)$

Proof

$$
\begin{aligned}
L(\theta|X) &= logp(X|\theta) \\
&= log \sum_{Z} P(X,Z|\theta) \\
&= log \sum_{Z} q(Z) \frac{P(X,Z|\theta)}{q(Z)} \\
&\geq \sum_{Z} q(Z) ln(\frac{P(X,Z|\theta)}{q(Z)}) \\
&= B(q,\theta)
\end{aligned}
$$

$L(\theta|X)$의 하한은 $B(q,\theta)$이므로, 이를 최대한 키워 L(θ|X)와 비슷하게 만든다

$\Rightarrow$ $B(q,\theta)$를 최대화 하자!

1.2 EM Algorithm using B function

EM Algorithm $L(\theta|X) = B(q,\theta) + KL(q \parallel p)$

1. E-step

L(θ|X) ≥ $B(q,\theta)$ 이기에, θ를 고정한 상태에서

$B(q,\theta)$를 최대한 키워 L(θ|X)와 비슷하게 만든다.

$B(q,\theta)$와 $KL(q \parallel p)$ 가 q에 대해 상보적 관계를 가짐

⇒ $KL(q \parallel p)$를 최소화하는 q를 선택하면,

이 q가 $B(q,\theta)$ 최대화한다.

⇒ $KL(q \parallel p) \geq 0$ 이므로 $KL(q \parallel p) = 0$,

즉 $q(Z) = p(Z|X,\theta)$이 되도록 q를 설정하면 $B(q,\theta)$

최대화할 수 있다

2. M-step

최대한 키운 $B(q,\theta)$를 최대화하는 θ를 찾는다.

3. θ가 수렴할 때까지 **E-step**과 **M-step**을 반복한다.

1. E-step

parameter를 고정하고 latent variable를 통계적으로 할당하는 과정

⇒ θ를 고정하고, $q(Z)$를 선택하는 과정

$$q^{new} = \underset{q}{argmax}\ B(q, \theta^{old})$$

2. M-step

할당한 결과를 바탕으로 parameter를 재산정하는 과정

⇒ 선택된 $q(Z)$를 가지고 θ를 다시 계산하는 과정

$$\theta^{new} = \underset{\theta}{argmax}\ B(q^{fixed}, \theta)$$

3. θ가 수렴할 때까지 **E-step**과 **M-step**을 반복한다.

1.2 EM Algorithm using B function

1. E-step

parameter를 고정하고 latent variable를 통계적으로 할당하는 과정

θ를 고정하고, $q(Z)$를 선택하는 과정

$$q^{new} = \underset{q}{argmax}\ B(q, \theta^{old})$$

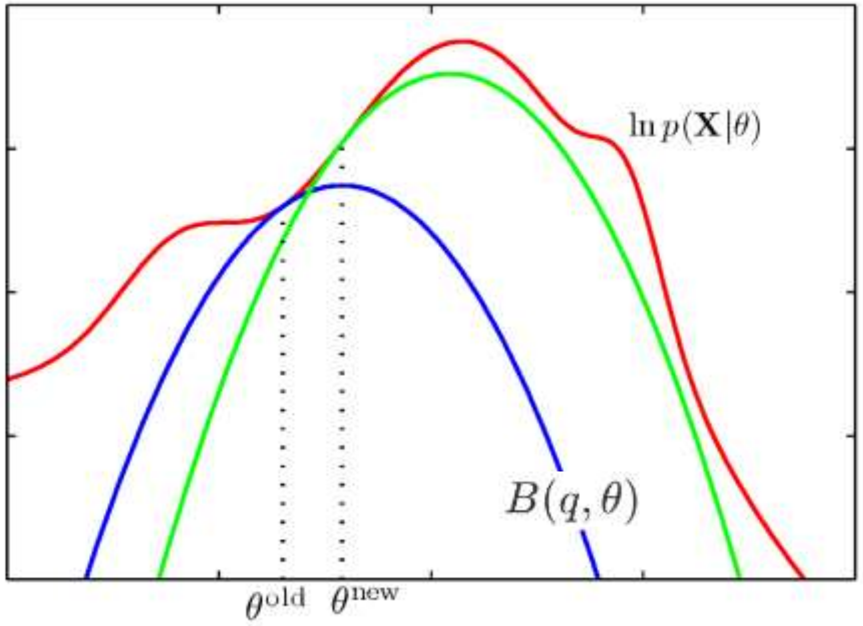
2. M-step

할당한 결과를 바탕으로 parameter를 재산정하는 과정

선택된 $q(Z)$를 가지고 θ를 다시 계산하는 과정

$$\theta^{new} = \underset{\theta}{argmax}\ B(q^{fixed}, \theta)$$

3. θ가 수렴할 때까지 **E-step**과 **M-step**을 반복한다.



1. E-step

θ를 θ^{old}로 고정한다. θ^{old}에서 $KL(q \parallel p) = 0$가 되도록,

즉 $L(X|\theta^{old}) = B(q, \theta^{old})$가 되도록 q를 선택하고 파란색 $B(q, \theta)$를 그린다.

2. M-step

파란색 $B(q, \theta)$를 최대화하는 새로운 파라미터 θ 값을 선정한다.

위의 그림에서는 θ^{new}가 되겠다.

3. θ가 수렴할 때까지 **E-step**과 **M-step** 반복

θ^{new}를 가지고 q를 선택하는 **E-step**를 반복한다. $L(X|\theta^{new}) = B(q, \theta^{new})$가 되도록 q를 선택하고 연두색 $B(q, \theta)$를 그린다. 이후 연두색 $B(q, \theta)$를 최대화하는 새로운 파라미터 θ 값을 선정한다.

1.3 EM Algorithm using Q function

Log-likelihood function L(θ|X)를 다른 방식으로 분해해보자.

$$L(\theta|X) = lnp(X|\theta)$$
$$= ln\frac{p(X,Z|\theta)}{p(Z|X,\theta)}$$
$$= lnp(X,Z|\theta) - lnp(Z|X,\theta)$$

$$L(\theta|X) = \sum_z L(\theta|X)p(\mathbf{Z}|\mathbf{X},\theta^{\mathbf{old}})$$
$$= \sum_{\mathbf{Z}} \ln p(\mathbf{X},\mathbf{Z}|\theta)p(\mathbf{Z}|\mathbf{X},\theta^{\mathbf{old}}) - \sum_z \ln p(\mathbf{Z}|X,\theta)p(\mathbf{Z}|\mathbf{X},\theta^{\mathbf{old}})$$
$$= Q(\theta|\theta^{old}) + H(\theta|\theta^{old})$$

$$L(\theta|X) = Q(\theta|\theta^{old}) + H(\theta|\theta^{old})$$

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

여기서 $H(\theta|\theta^{old})$는 음수의 합**(negated sum)**으로 정의된다.

$$L(\theta|X) - L(\theta^{old}|X) \geq Q(\theta|\theta^{old}) - Q(\theta^{old}|\theta^{old})$$

Proof)
$$L(\theta|X) = Q(\theta|\theta^{old}) + H(\theta|\theta^{old})$$
$$L(\theta^{old}|X) = Q(\theta^{old}|\theta^{old}) + H(\theta^{old}|\theta^{old})$$
$$L(\theta|X) - L(\theta^{old}|X) = Q(\theta|\theta^{old}) - Q(\theta^{old}|\theta^{old}) + H(\theta|\theta^{old}) - H(\theta^{old}|\theta^{old})$$

From Gibb's inequality, $H(\theta|\theta^{old}) \geq H(\theta^{old}|\theta^{old})$

$$\therefore\ L(\theta|X) - L(\theta^{old}|X) \geq Q(\theta|\theta^{old}) - Q(\theta^{old}|\theta^{old})$$

1.3 EM Algorithm using Q function

$$L(\theta|X) = Q(\theta|\theta^{old}) + H(\theta|\theta^{old})$$

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

여기서 $H(\theta|\theta^{old})$는 음수의 합**(negated sum)**으로 정의된다.

$$L(\theta|X) - L(\theta^{old}|X) \geq Q(\theta|\theta^{old}) - Q(\theta^{old}|\theta^{old})$$

$Q(\theta|\theta^{old})$을 향상시키는 θ 를 찾으면 $L(\theta|X)$가 향상된다.

즉, $Q(\theta|\theta^{old})$를 최대화하는 θ는 **log-likelihood** $L(\theta|X)$의 값도 증가시킨다.

⇒ 결론 : $Q(\theta|\theta^{old})$ 를 목적함수로 사용 가능하다!

- **EM Algorithm**

1. **E-step**

$Q(\theta|\theta^{old})$를 계산($\ln p(X, Z|\theta)$의 기댓값 계산)

2. **M-step**

$$\theta^{new} = \underset{\theta}{argmax}\ Q(\theta|\theta^{old})$$

- $B(q,\theta)$와 $Q(\theta|\theta^{old})$의 관계를 알아보자.

$$B(q,\theta)$$

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

$$= Q(\theta|\theta^{old}) + constant$$

⇒ $B(q,\theta)$를 최대화하는 과정은 $Q(\theta|\theta^{old})$를 최대화하는 과정과 같다!

1.4 Example with R code

Example : 한 혈액형에 대해 n_+, n_-가 주어졌을 때, +형일 확률 $\theta = \frac{n_+}{n_+ + n_-}$를 추정해보자!

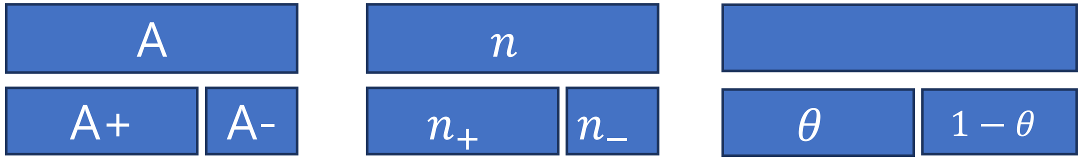
<Notation>

n : 한 혈액형의 인원 수

n_+ : +형인 인원 수 → 관찰 데이터 y_{obs} : (n_+, n_-)

$n_- = n_{--}$: -형인 인원 수 ↗

θ : +형의 비율 $\theta = \frac{n_+}{n_+ + n_-}$



$$n_+ \sim Binomial(n, \theta)$$

가장 먼저 EM Algorithm의 목표가 되는 θ 의 MLE를 구해보자

혈액형	A	B	O	AB
n_+ : +형 인원 수	4,159,001,000명	3,283,570,000명	5,748,581,000명	987,724,000명
n_- : -형 인원 수	314,825,700명	173,151,000명	403,702,300명	58,034,200명
$\hat{\theta}$: MLE of θ	0.9296058	0.9499439	0.9343666	0.9445185

$$\hat{\theta} = \frac{n_+}{n}$$

$y \sim Binomial(n, \theta)$ 일 때, θ의 MLE $\hat{\theta} = \frac{y}{n}$인 점을 활용하여

$n_+ \sim Binomial(n, \theta)$ 에서 θ의 MLE $\hat{\theta} = \frac{n_+}{n}$이다.

1.4 Example with R code

EM Algorithm을 적용하기 위하여 Latent variable를 정의한다.

<추가 Notation>

n_{++} : +형인 사람 중 ++인 인원 수

n_{+-} : +형인 사람 중 +-인 인원 수

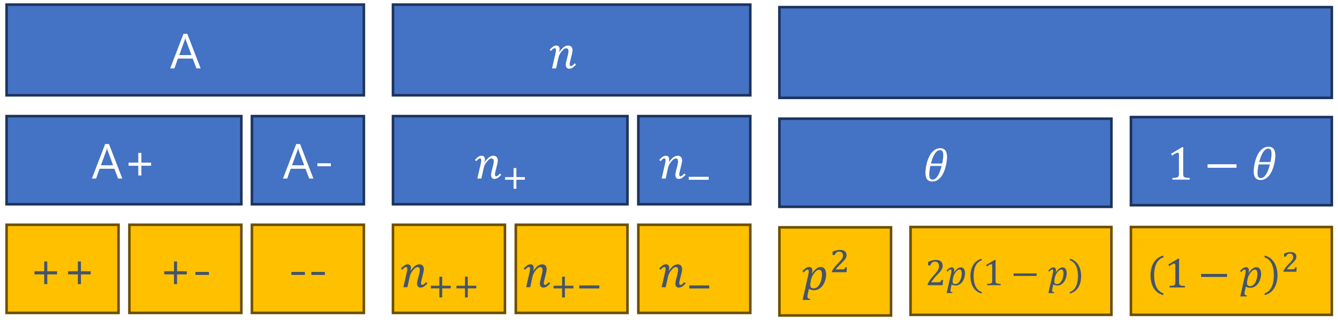
→ Latent Variable $Z = (n_{++}, n_{+-})$

p : (+) 요소를 가질 확률 $\theta = p^2 + 2p(1-p) = 2p - p^2$

y_{obs} : 관찰데이터 (n_+, n_-)

y_{com} : 관찰데이터와 latent variable

$(n_+, n_-, n_{++}, n_{+-})$



혈액형	A	B	O	AB
$\hat{\theta}$: MLE of θ	0.9296058	0.9499439	0.9343666	0.9445185
p 수렴 값	?	?	?	?
반복 수	?	?	?	?
θ 수렴 값	?	?	?	?

p에 대한 EM Algorithm으로 p의 수렴 값을 구하고, θ와 p의 관계식을 이용해 최종 θ의 수렴 값을 구한다.

1.4 Example with R code

A		
A+		A-
++	+-	--

n		
n_+		n_-
n_{++}	n_{+-}	n_-

θ		$1-\theta$
p^2	$2p(1-p)$	$(1-p)^2$

$$P(p|y_{com}) \propto (p^2)^{n_{++}}(2p(1-p))^{n_{+-}}(1-p)^{2n_-}$$
$$L(p|y_{com}) = n_{++}\log(p^2) + n_{+-}\log(2p(1-p)) + 2n_-\log(1-p) + constant$$

1. **E-step** : $Q(p|p^{old})$ 계산

$$Q(p|p^{old}) = E[L(p|y_{com})|y_{obs}, p^{old}]$$
$$= E[n_{++}|y_{obs}, p^{old}]log(p^2) + E[n_{+-}|y_{obs}, p^{old}]log(2p(1-p)) + n_{-}log(1-p)^2 + C$$

1.4 Example with R code

1. **E-step** : Q($p|p^{old}$) 계산

$$Q(p|p^{old}) = E[L(p|y_{com})|y_{obs}, p^{old}]$$
$$= E[n_{++}|y_{obs}, p^{old}]log(p^2) + E[n_{+-}|y_{obs}, p^{old}]log(2p(1-p)) + n_{--}log(1-p)^2 + C$$

- 기댓값을 구하기 위해 Multinomial 분포의 성질을 이용한다.

$$(y_1, ..., y_k) \sim Multinomial(n; p_1, ..., p_k)$$
$$\therefore y_i|y_i + y_j \sim Binomial(y_i + y_j, \frac{p_i}{p_i + p_j})$$

을 그대로 적용하여 ($n_{++}|y_{obs}, p^{old}$) 의 분포를 구한 결과 아래와 같다

$$n_{++}, n_{+-}, n_{--} \sim Multinomial(n; p^2, 2p(1-p), (1-p)^2)$$
$$\therefore n_{++}|n_{++} + n_{+-}, p^{old} \sim Binomial(n_+; \frac{(p^{old})^2}{(p^{old})^2 + 2p^{old}(1-p^{old})})$$

- 방금 구한 조건부 분포로 기댓값을 구한다.

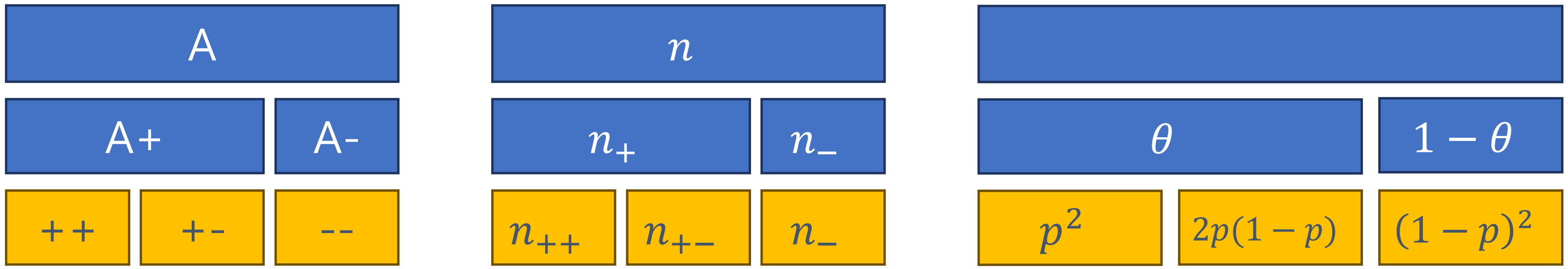
$$\widehat{n_{++}} = E[n_{++}|y_{obs}, p^{old}] = (n_{++} + n_{+-})\frac{(p^{old})^2}{(p^{old})^2 + 2p^{old}(1-p^{old})}$$

$$\begin{aligned}\widehat{n_{+-}} &= E[n_{+-}|y_{obs}, p^{old}] \\ &= E[n_+ - n_{++}|y_{obs}, p^{old}] \\ &= n_+ - E[n_{++}|y_{obs}, p^{old}] \\ &= n_+ - \widehat{n_{++}}\end{aligned}$$

- 기댓값을 대입하여 **E-step**에서 계산된 Q($p|p^{old}$)은 다음과 같다.

$$Q(p|p^{old}) = \widehat{n_{++}}\log(p^2) + \widehat{n_{+-}}\log(2p(1-p)) + 2n_{-}\log(1-p) + constant$$

1.4 Example with R code



2. M-step : $\widehat{n_{++}}, \widehat{n_{+-}}$ 일 때, $Q(p|p^{old})$ 최대화하는 **parameter** p 찾기

$$Q(p|p^{old}) = \widehat{n_{++}}\log(p^2) + \widehat{n_{+-}}\log(2p(1-p)) + 2n_-\log(1-p) + constant$$

$$\frac{\partial Q(p|p^{old})}{\partial p} = \widehat{n_{++}}\frac{2}{p} + \widehat{n_{+-}}\frac{1-2p}{p(1-p)} + n_-\frac{2}{p-1} \overset{\text{set}}{=} 0$$

$$p^{new} = \frac{2\widehat{n_{++}} + \widehat{n_{+-}}}{2(\widehat{n_{++}} + \widehat{n_{+-}} + n_-)}$$

1.4 Example with R code

```r
EM <- function(p, y){

  #초깃값
  iter<-0
  value<-NULL

  #repeat E and M steps
  repeat{
    ##E step
    npp<-y[1]*(p^2)/(p^2+2*p*(1-p))
    npn<-y[1]- npp

    ##M step
    p_new <- (2*npp+npn)/(2*(npp+npn+y[2]))

    ## update old to new
    iter<- iter+1
    value[iter]<-p_new
    epsilon<-abs(p_new-p)
    p<- p_new



    ##decision based on
    if(epsilon <0.0000001){break}

  }
  list(p=p_new,
       step=iter,
       whole=value)
}
```

y = c(n_+, n_-)

1. **E-step** : $Q(p|p^{old})$ 계산

$$\widehat{n_{++}} = E[n_{++}|y_{obs}, p^{old}] = (n_{++} + n_{+-})\frac{(p^{old})^2}{(p^{old})^2 + 2p^{old}(1-p^{old})}$$

$$\widehat{n_{+-}} = n_{+} - \widehat{n_{++}}$$

2. **M-step** : $\widehat{n_{++}}, \widehat{n_{+-}}$ 일 때, $Q(p|p^{old})$

최대화하는 **parameter** p 찾기

$$p^{new} = \frac{2\widehat{n_{++}} + \widehat{n_{+-}}}{2(\widehat{n_{++}} + \widehat{n_{+-}} + n_{-})}$$

3. θ가 수렴할 때까지 **E-step**과 **M-step**을 반복한다.

1.4 Example with R code

혈액형	A	B	O	AB
n_+ : +형 인원 수	4,159,001,000명	3,283,570,000명	5,748,581,000명	987,724,000명
n_- : -형 인원 수	314,825,700명	173,151,000명	403,702,300명	58,034,200명
$\hat{\theta}$: MLE of θ	0.9296058	0.9499439	0.9343666	0.9445185
p 수렴 값	0.7347254	0.7761894	0.7761894	0.7644264
반복 수	26	31	27	29
θ 수렴 값	0.9296294	0.9499088	0.9343816	0.9445051

```
> A_world=c(4159001000,314825700)
> EM(0.5,A_world)
$p
[1] 0.7347254

$step
[1] 26

$whole
 [1] 0.6197530 0.6735239 0.7008264 0.7155544 0.7237593
 [6] 0.7284123 0.7310777 0.7326134 0.7335011 0.7340152
[11] 0.7343133 0.7344862 0.7345866 0.7346448 0.7346787
[16] 0.7346983 0.7347097 0.7347163 0.7347202 0.7347224
[21] 0.7347237 0.7347245 0.7347249 0.7347251 0.7347253
[26] 0.7347254
```

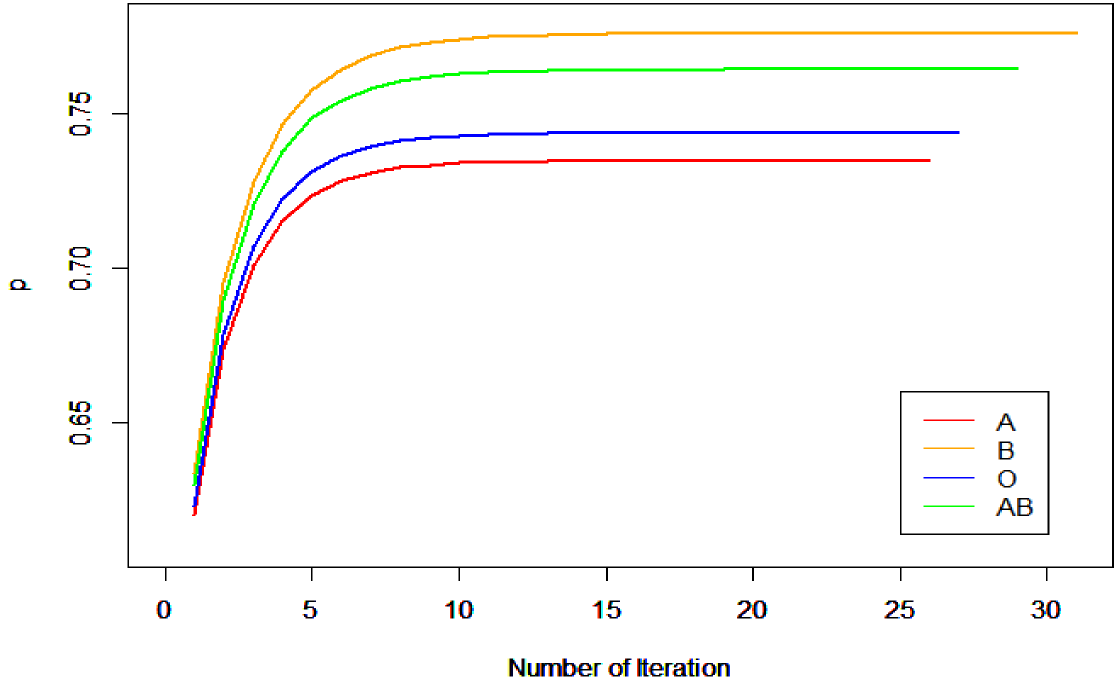
A형 데이터로 EM Algorithm을 돌린 결과

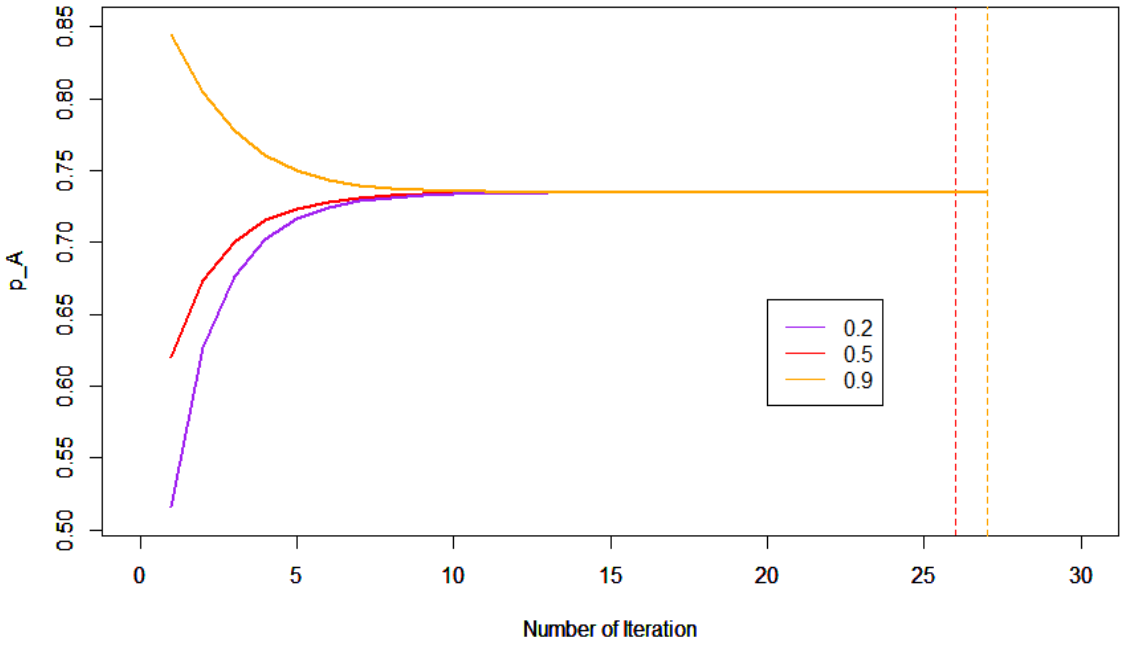
p에 대한 EM Algorithm으로 p의 수렴 값을 구하고,

θ와 p의 관계식을 이용해 우리가 원하는 모수인 θ의 수렴 값을 구한다.

$$\theta = p^2 + 2p(1-p) = 2p - p^2$$

1.4 Example with R code





0.5의 초깃값으로 A형, B형, O형, AB형 EM Algorithm을 돌린 결과

⇒ 각각 26, 31, 27, 29번의 반복을 통해 p가 수렴했다.

⇒ B형 > AB형 > O형 > A형의 순서로 +의 비율(θ)이 높다

A형 데이터로, 초깃값을 다르게 하여 EM Algorithm을 돌린 결과

⇒ 초반의 값이 다르지만, 모두 같은 값으로 수렴했다

2

EM Algorithm with missing data

2.1 Condition for EM Algorithm with missing data

결측치가 있을 때의 EM 알고리즘의 개념

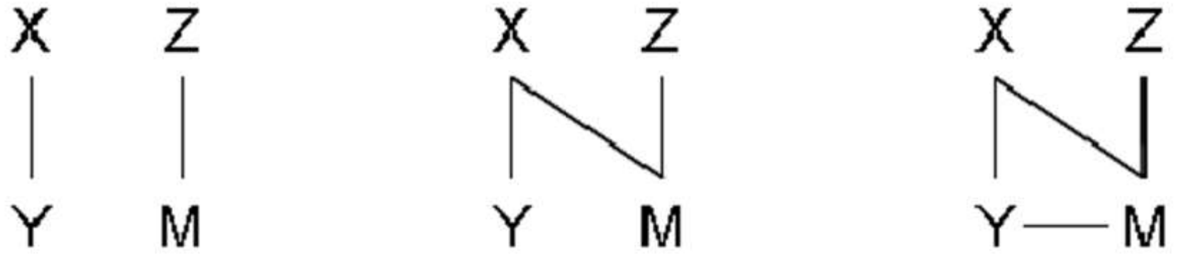
E-step: **관측된 데이터의 충분 통계량(sufficient statistics) 을 이용해 결측 데이터 (missing data) 를 채운다**.

(관측된 데이터와 현재 파라미터 θ^t를 기반으로 결측 데이터의 조건부 기댓값을 계산한다.)

M-step: E 단계에서 계산된 통계량을 사용해 로그우도함수를 최대화하고 θ를 최적화한다.

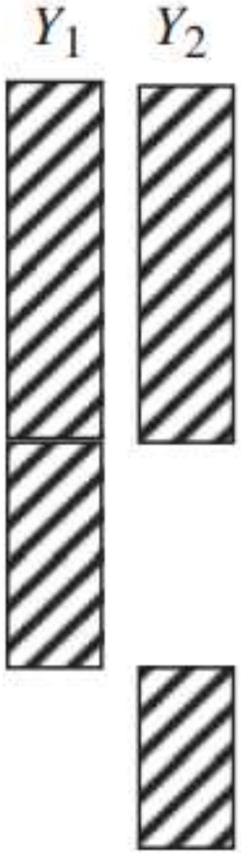
이때 Complete Data에서 MLE를 사용하는 것과 비슷하다.

2.1 Condition for EM Algorithm with Missing Data



(a) MCAR (b) MAR (c) MNAR

우리가 다룬 세 missing data 타입 중
어떤 경우에 EM 알고리즘을 적용할 수 있을까?



Y2의 결측치를 Y2의 관측 데이터에서 추정 가능하려면,
결측치가 종속변수의 관측된 데이터 & 독립변수와 모두
관련이 없어야 한다. → **MCAR**

2.2 Mathematical Explanation of EM with missing data

$Y_i \sim N(\mu, \sigma^2)$ 이고 Y_i가 $i = 1, \dots, r$ 까지는 관측되고 $i = r+1, \dots, n$ 까지는 관측되지 않는 상황

E-step:

$$E\left(\sum_{i=1}^{n} y_i \mid \theta^{(t)}, Y_{(0)}\right) = \sum_{i=1}^{r} y_i + (n-r)\mu^{(t)}$$

데이터 합의 조건부 기대값

1~r까지의 r개의 관측된 데이터의 합

r+1~n까지, (n-r)개 결측된 데이터의 기대합

and

$$E\left(\sum_{i=1}^{n} y_i^2 \mid \theta^{(t)}, Y_{(0)}\right) = \sum_{i=1}^{r} y_i^2 + (n-r)\left[\left(\mu^{(t)}\right)^2 + \left(\sigma^{(t)}\right)^2\right],$$

데이터 제곱합의 조건부 기대값

관측된 데이터의 제곱합

(n-r)개 결측된 데이터 제곱의 기대합: 결측된 데이터가 **분산에 미치는 영향도** 반영하기 위해

2.2 Mathematical Explanation of EM with missing data

M-step

$$\mu^{(t+1)} = E\left(\sum_{i=1}^{n} y_i \mid \theta^{(t)}, Y_{(0)}\right) \Big/ n,$$

μ^t 다음으로 추정되는 μ

앞에서 구한 데이터 합의 조건부 기댓값

$$(\sigma^{(t+1)})^2 = E\left(\sum_{i=1}^{n} y_i^2 \mid \theta^{(t)}, Y_{(0)}\right) \Big/ n - (\mu^{(t+1)})^2.$$

σ^{t+1}도 비슷한 방식으로 추정. 이때 $V(X) = E(X^2)$-$\{E(X)\}^2$ 간편식 사용

$\mu^{(t)} = \mu^{(t+1)} = \hat{\mu}$ and $\sigma^{(t)} = \sigma^{(t+1)} = \hat{\sigma}$

될 때까지 E, M 스텝 iteration 수행. 수행을 멈추는 지점을 fixed point 라고 부름

2.3 Example 1 (Simple)

$Y_1 \sim N(\mu, \sigma^2)$ 이고 Y_1가 $i = 1, \ldots., 6$ 까지는 관측되고 $i = 7, \ldots, 10$ 까지는 관측되지 않은 상황

관측된 데이터의 합: 30 , $\mu = 7$.

결측된 데이터 4개도 $\mu = 7$ 을 따른다고 가정한다.

E step: 데이터 합의 조건부 기대값 = 관측된 데이터의 합 + 결측된 데이터합의 기댓값 = 30 + 7 x 4 = 58

M step: 58 $\div 10 = 5.8 = \mu^{t+1}$

새로운 μ^{t+1}를 결측된 데이터의 평균으로 사용하고 E,M 스텝을 반복한다.

E step: 30 + 5.8 x 4 = 53.2

M step: 53.2 $\div 10 = 5.32$

…. 특정 값으로 수렴할 때 까지 반복하여 최적의 μ를 찾는다.

2.3 Example 2 (From Textbook)

관측된 데이터 $Y_0 = (38,34,125)$가 다음과 같은 다항분포 $(\frac{1}{2}-\frac{\theta}{2}, \frac{1}{2}+\frac{\theta}{4})$ 를 따르고, 확장된 데이터셋 $Y = (y_1, y_2, y_3, y_4)$은 다항분포 $(\frac{1}{2}-\frac{\theta}{2}, \frac{\theta}{4}, \frac{\theta}{4}, \frac{1}{2})$ 를 따른다. Y의 일부가 관측되지 않았기 때문에 EM 알고리즘을 사용해 θ를 추정하고자 하는 상황.

$$E(y_1 \mid \theta, Y_{(0)}) = 38,$$
$$E(y_2 \mid \theta, Y_{(0)}) = 34,$$
$$E(y_3 \mid \theta, Y_{(0)}) = 125(\theta/4)/(1/2 + \theta/4),$$
$$E(y_4 \mid \theta, Y_{(0)}) = 125(1/2)/(1/2 + \theta/4).$$

$$\theta^{(t+1)} = \left(34 + y_3^{(t)}\right) / \left(72 + y_3^{(t)}\right).$$

Rate of convergence

t	$\theta^{(t)}$	$\theta^{(t)} - \hat{\theta}$	$(\theta^{(t+1)} - \hat{\theta})/(\theta^{(t)} - \hat{\theta})$
0	0.500 000 000	0.126 821 498	0.146 5
1	0.608 247 423	0.018 574 075	0.134 6
2	0.624 321 051	0.002 500 447	0.133 0
3	0.626 488 879	0.000 332 619	0.132 8
4	0.626 777 323	0.000 044 176	0.132 8
5	0.626 815 632	0.000 005 866	0.132 8
6	0.626 820 719	0.000 000 779	
7	0.626 821 395	0.000 000 104	
8	0.626 821 484	0.000 000 014	

2.3 Example 2 (From Textbook)

이 식이 왜 rate of convergence (수렴 속도)를 나타내는가?

$$\frac{\theta^{(t+1)}-\hat{\theta}}{\theta^{(t)}-\hat{\theta}}$$

다음 반복에서의 오차

현재 반복에서의 오차

현재에서 다음으로 업데이트 되었을 때의 오차 비율 → 수렴 속도의 개념

$\hat{\theta}$: 찾고자 하는 최적의 parameter

$\|\theta^{(t+1)}-\hat{\theta}\| \leq c\|\theta^{(t)}-\hat{\theta}\|$ 에서 0과 1사이의 상수 c 가 일정한 값으로 수렴한다면 선형 수렴(linear convergence)를 나타냄

2.3 Example 3 (Python)

$n = 6$ 인 데이터셋 Y에서 다음과 같이 3개의 값은 관측되고 나머지 3개의 값은 관측되지 않았다고 하자.

Y
10
5
1
?
?
?

그럼 우리가 원래 하던 방식대로, E-step은 결측된 값들이 관측된 값들의 평균을 따른다고 가정하고 값을 대입하면 된다.

```
X=np.array([[10,1,5]])
N_miss=3
Mean=np.sum(x)/(n+n_miss)
```

을 수행하면 mean=2.66을 얻는다.

Y
10
5
1
2.66
2.66
2.66

2.3 Example 3 (Python)

M 스텝에서 얻어진 최적의 $\hat{\mu}$ 를 결측치에 대입하고, 그 상태의 데이터셋에서 MLE를 적용해서 최적의 θ를 찾는다.

```
Prev_mean=0
Updated mean = mean + (missingvalues + n_miss) / (n + n_miss)
Missingvalues = updated_mean
Mean_difference = updated mean - prev_mean
Print('\n The Current mean is: )
Print('\n The mean Difference is: )
If(mean_difference < 0.05):
       break
```

```
The Current mean is : 4.167
The mean Difference is 4.167
The Current mean is : 4.75
The mean Difference is : 0.583
…..
The Current mean is : 5.297
The mean Difference is : 0.036
```

Y
10
5
1
5.29
5.29
5.29

2.3 Example 3 (Python)

M 스텝에서 얻어진 최적의 $\hat{\mu}$ 를 결측치에 대입하고, 그 상태의 데이터셋에서 MLE를 적용해서 최적의 θ를 찾는다.

```
For i in range (n_miss):
      x = np.append(x,np.array([[updated_mean]]),axis = 1)
Print (x)
```

```
[[10.    1.       5.       5.2968       5.2968       5.2968]]
```

```
Making Sample Parameter sets:
New_mean=np.array([[2, 5, 1, 3]])
New_sigma=np.array([[4,2,1,6]])
```

가우시안 분포이므로 μ, σ의 2개로 이루어진 샘플 파라미터 셋을 구성한다.

2.3 Example 3 (Python)

M 스텝에서 얻어진 최적의 $\hat{\mu}$ 를 결측치에 대입하고, 그 상태의 데이터셋에서 MLE를 적용해서 최적의 θ를 찾는다.

$$Log\ [Likelihood(x|\theta)] = \log\left(\prod_{i=1}^{n}\frac{1}{\sqrt{2\pi}\sigma}e^{\frac{-(x_i-\mu)^2}{2\sigma^2}}\right)$$

Log를 씌워 범위를 좁혀준다(어차피 최대가 되는 θ가 무엇인지만 판단하면 되니까)

$$Log\ [Likelihood(x|\theta)] = \sum_{i=1}^{n}\frac{(x_i-\mu)^2}{2\sigma^2} -- 0.5 \text{ x } \log 2\pi - n * \log\sigma$$

```
Log_likelihood = -np.sum(np.square(x – new_mean[0 , i]) / 2 * np.square(sigma[0, i]))) – 0.5* n*
np.log10(2* math.pi))-n* np.log10(sigma[0, i])
Print(new_mean[0, i], new_sigma[0, i], log_likelihood)
_________________________________________-
2   4    -9.338
5    2     -9.359
1   1    -78.589
3    6     -8.075 (MAX)
```

Sample parameter 중에서 Log likelihood가 최대가 되는 조합인 (3,6)을 optimal parameter로 결정할 수 있다.

2.4 Exponential Family and EM

데이터의 분포가 지수족(Exponential Family)일 때는 EM 알고리즘을 수행하기에 더 쉽고 명확하다.

지수족: pdf가 다음의 형태로 표현되는 분포들.

예) 베르누이 분포, 정규분포

$$f(Y \mid \theta) = b(Y)\exp(s(Y)\theta - a(\theta))$$

$s(Y)$ = 완전한 데이터에서의 충분 통계량

a, b = 각각 θ, Y 의 함수

$$f(Y \mid \theta) = b(Y)\exp(s(Y)\theta - a(\theta)) \xrightarrow{\text{로그}} \log f(y|\theta) = \log b(Y) + s(Y)\theta - a\theta$$

→ 지수족에서는 충분 통계량 $s(Y)$ 와 θ가 선형 결합으로 나타나게 된다. 이는 로그 우도함수가 $s(Y)$ 와 θ에 대해 선형 관계를 가진다는 것을 의미한다.

따라서 E 스텝에서 계산이 간단해지는 장점을 갖는다. M 단계에서도 로그 우도함수의 최적화가 더 쉬워진다.

2.5 EM Extensions _GEM

일반화된 EM 알고리즘 (Generalized EM)

E 단계는 EM 알고리즘과 같지만, M 단계에서 바로 우도함수를 최대화할 필요 없이 그저 우도함수를 증가시키는 방향으로 업데이트가 이루어지기만 하면 된다.

Theorem 8.1 *Every GEM algorithm increases $\ell(\theta \mid Y_{(0)})$ at each iteration, that is,*

$$\ell\left(\theta^{(t+1)} \mid Y_{(0)}\right) \geq \ell\left(\theta^{(t)} \mid Y_{(0)}\right),$$

with equality if and only if

$$Q(\theta^{(t+1)} \mid \theta^{(t)}) = Q(\theta^{(t)} \mid \theta^{(t)}).$$

더 이상 커질 수 없을 때 멈춤
→ 수렴은 보장된다.

최대화를 바로 수행하기 어려운 복잡한 모델에서의 사용에 적합하다.

크고 복잡한 모델이나 데이터셋에 대해 작은 부분적으로 접근해 점진적으로 개선한다는 점에서 SGD (확률적 경사하강법)과 유사점을 갖는다.

2.5 EM Extensions _ECM

ECM 알고리즘 (Expectaion Conditional – Maximization) 은 조건부 최대화의 개념을 도입한다. 이는 하나의 파라미터를 최대화할 때 나머지 파라미터를 고정하는 방식이다. 파라미터를 한번에 최대화하는 것이 아니라 그룹으로 묶어 그룹 단위로 최대화 과정을 수행하는 것이다.

$$\theta_1^{(t+1)} = \arg\max_{\theta_1} Q(\theta_1, \theta_2^{(t)}, \ldots, \theta_m^{(t)} | \theta^{(t)})$$
$$\theta_2^{(t+1)} = \arg\max_{\theta_2} Q(\theta_1^{(t+1)}, \theta_2, \ldots, \theta_m^{(t)} | \theta^{(t)})$$
$$\cdots$$
$$\theta_m^{(t+1)} = \arg\max_{\theta_m} Q(\theta_1^{(t+1)}, \theta_2^{(t+1)}, \ldots, \theta_m | \theta^{(t)})$$

각각의 조건부 최대화 단계에서 Q(missing data의 조건부 기댓값) 을 순차적으로 증가 (monotonically increase) 해서 각 파라미터를 순차적으로 최대한다는 점에서 GEM 알고리즘과 비슷하다고 할 수 있다.

감사합니다