

확률통계 및 공통 알고리즘

6. 공통 알고리즘 -I

임채영

서울대학교

이번 강의에서 다룰 내용

- ▶ 함수의 해 찾기
- ▶ 기초적인 최적화 알고리즘
- ▶ MM 알고리즘
- ▶ EM 알고리즘
- ▶ 몬테카를로 방법 (Monte Carlo method)
- ▶ 변분추론 (Variational inference) -ELBO 최적화

최적화 (optimization) 방법들

- ▶ 통계분석 방법이나 기계학습 방법등 많은 데이터 분석 방법에서는 목적함수(objective function)가 최소 또는 최대가 되도록 하는 모수를 찾는것이 필요하다.
- ▶ 손실함수인 경우 최소가 되도록 하고 가능도 함수인 경우는 최대가 되게 한다.

예: 최소제곱법, 최대가능도법 등

- ▶ 함수의 최대 또는 최소가 되는 해를 찾는 알고리즘들을 최적화 알고리즘 (optimization algorithm)이라고 한다.

함수의 해 찾기 (Finding a root)

- ▶ 목적함수가 미분가능한 경우 목적함수의 최대(최소)를 찾는것은 미분한 함수의 해를 찾는 문제와 같다.
- ▶ 즉, $f(x) = 0$ 을 만족시키는 x 를 찾는 문제
- ▶ 함수의 해를 찾는 알고리즘들: 이분법(Bisection method), 고정점 반복법 (Fixed point method), 뉴턴-랩슨 알고리즘 (Newton-Raphson method), 할선법 (Secant method)등

뉴턴-랩슨 방법

- ▶ x^* 를 $f(x)$ 의 해 (root)라고 하자. 즉, $f(x^*) = 0$
- ▶ N-R 알고리즘은 테일러 전개를 통해 유도할수 있다.

$$0 = f(x^*) \approx f(x_n) + f'(x_n)(x^* - x_n)$$

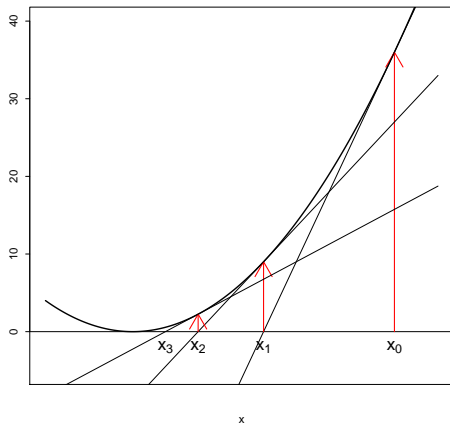
- ▶ $x^* \approx x_n - \frac{f(x_n)}{f'(x_n)}$

- ▶ 따라서,

$$x_{new} = x_{old} - \frac{f(x_{old})}{f'(x_{old})}$$

기하학적 의미:

$$0 = f'(x_{old})(x_{new} - x_{old}) + f(x_{old})$$



할선법

- ▶ N-R 방법의 변형
- ▶ $f'(x)$ 를 수치적으로 근사하여 대입.
- ▶ 즉, $f'(x_r) \approx \frac{f(x_r) - f(x_{r-1})}{x_r - x_{r-1}}$ 을 이용.
- ▶
$$x_{r+1} = x_r - \frac{x_r - x_{r-1}}{f(x_r) - f(x_{r-1})} f(x_r)$$
- ▶ cf. N-R: $x_{r+1} = x_r - f(x_r)/f'(x_r)$
- ▶ 할선법은 다음의 경우 유용하다.
 - ▶ $f'(x)$ 를 구하기 어려울때,
 - ▶ f' 를 계산하는것이 f 를 계산하는것 보다 계산의 복잡도가 더 클때

최적화(Optimization)

- ▶ $\mathbf{x}_{opt} = \arg \min_{\mathbf{x}} (\text{or } \arg \max) F(\mathbf{x}), \mathbf{x} \in R^p.$
- ▶ 최대를 찾는 문제와 최소를 찾는 문제는 같은 방식으로 해결 가능 ($F(\mathbf{x})$ 를 $-F(\mathbf{x})$ 로 바꾸기)
- ▶ F' 을 쉽게 구할 수 있으면 함수의 해를 찾는 문제, 즉 $F'(x) = 0$ 을 찾는 알고리즘들을 사용할 수 있음.
- ▶ 아니면 다른 최적화 알고리즘 사용: 경사하강법 (Gradient descent method) 등

다변량 뉴턴-랩슨 방법

- ▶ gradient vector (기울기 벡터)
 $\nabla F(\mathbf{x}) = (dF(\mathbf{x})/dx_1, \dots, dF(\mathbf{x})/dx_p)$ 를 구하기 쉬우면,
최적화 문제를 $\nabla F(\mathbf{x}) = 0$ 의 해를 찾는 문제로 본다.
- ▶ 다변량 함수의 테일러 전개를 이용 다변량 뉴턴-랩슨 방법을 유도할 수 있다.

$$\mathbf{x}_{new} = \mathbf{x}_{old} - \mathbf{H}(\mathbf{x}_{old})^{-1} \nabla F(\mathbf{x}_{old}),$$

$\mathbf{H}(\mathbf{x}) = \nabla^2 F(\mathbf{x})$ 는 두번 미분한 함수들로 이루어진
헤시안행렬 (Hessian matrix). $\mathbf{H}_{ij} = d^2 F(\mathbf{x})/dx_i dx_j$.

- ▶ $\mathbf{H}(\mathbf{x})$ 이 $\nabla F(\mathbf{x}) = 0$ 의 해 (stationary point)에서 양의 정부호 (positive definite) 일때, 국소 최소값(local minimum)을 찾을 수 있다.

경사하강법 (Gradient descent method)

- ▶ gradient vector $\nabla F(\mathbf{x}) = (dF(\mathbf{x})/dx_1, \dots, dF(\mathbf{x})/dx_p)$ 는 목적함수 $y = F(\mathbf{x})$ 의 기울기벡터로 F 값이 \mathbf{x} 에서 가장 가파르게 증가하는 방향을 나타낸다.
- ▶ 따라서, 기존 위치 \mathbf{x}_{old} 에서 가장 가파르게 감소하는 방향 $(-\nabla F(\mathbf{x}_{old}))$ 으로 α 만큼 업데이트 하도록 하는 알고리즘을 경사하강법 (Gradient Descent method, GD method)이라고 한다 (최소가 되는 \mathbf{x}_* 를 찾는 경우). 즉,

$$\mathbf{x}_{new} = \mathbf{x}_{old} - \alpha \nabla F(\mathbf{x}_{old}),$$

경사하강법의 이해

- ▶ 테일러 전개를 통한 2차함수 근사값을 최소화 하는 해를 찾는 문제로 볼 수 있다.

$$\begin{aligned} F(\mathbf{x}_{new}) \approx & F(\mathbf{x}_{old}) + \nabla F(\mathbf{x}_{old})^T (\mathbf{x}_{new} - \mathbf{x}_{old}) \\ & + \frac{1}{2} (\mathbf{x}_{new} - \mathbf{x}_{old})^T \nabla^2 F(\mathbf{x}_{old}) (\mathbf{x}_{new} - \mathbf{x}_{old}) \end{aligned}$$

- ▶ 여기서 $\nabla^2 F(\mathbf{x}_{old}) = \frac{1}{\alpha}$ 로 바꾸고, 근사된 2차함수식을 최소가 되게하는 위치 \mathbf{x}_{new} 를 찾으면 $\mathbf{x}_{new} = \mathbf{x}_{old} - \alpha \nabla F(\mathbf{x}_{old})$, 이 나온다.

최적화 문제에서 N-R 방법과 GD 방법의 차이

▶ $\operatorname{argmin}_{\mathbf{x}} F(\mathbf{x})$ 에서

- N-R은 $\nabla F(\mathbf{x}) = 0$ 의 해를 찾는 방법으로 접근
- GD는 목적함수가 가장 가파르게 감소하는 방향으로 업데이트 하는 방법

경사하강법의 변형

- ▶ 스텝사이즈 (step size) α 와 하강 방향을 조정함에 따라 다양한 변형된 방법이 존재한다.
- ▶ 주어진 목적함수의 형태에 따라 다양한 변형된 방법이 존재한다.

Stochastic GD method

- ▶ 목적함수가 $F(\mathbf{x}) = \sum_{i=1}^n F_i(\mathbf{x})$ 의 형태인 경우 (예: Likelihood, n 은 데이터 사이즈)
- ▶ $\nabla F(\mathbf{x}) = \sum_{i=1}^n \nabla F_i(\mathbf{x})$ 이므로

$$\mathbf{x}_{new} = \mathbf{x}_{old} - \alpha \sum_{i=1}^n \nabla F_i(\mathbf{x}_{old})$$

- ▶ n 이 클 경우 $\nabla F_i(\mathbf{x})$ 의 계산에 시간이 오래 걸릴 수 있다.
- ▶ SGD는 전체 gradient를 계산하지 않고 하나만 계산을 하여 업데이트 한다.

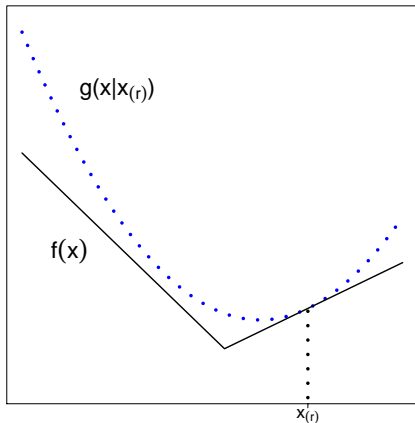
$$\mathbf{x}_{new} = \mathbf{x}_{old} - \alpha \nabla F_i(\mathbf{x}_{old})$$

- ▶ i 는 $i = 1, \dots, n$ 으로 차례로 선택하거나 랜덤하게 선택한다.

MM 알고리즘

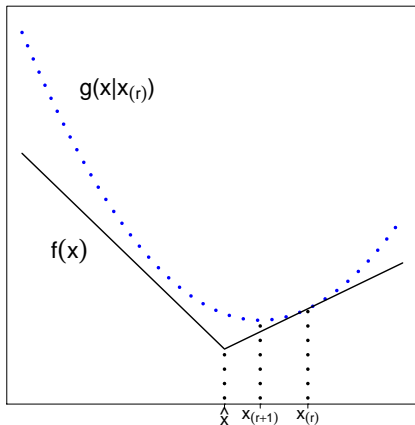
- ▶ MM 알고리즘은 볼록함수의 최소화 문제에서 대체 함수를 이용한 방법이다.
- ▶ EM 알고리즘은 MM 알고리즘의 특별 케이스로 볼 수 있다.
- ▶ MM은 최소화 문제에서는 Majorize-Minimize를 뜻하고, 최대화 문제에서는 Minorize-Maximize를 뜻한다.

- ▶ 최소화 문제 $\operatorname{argmin}_{\mathbf{x}} f(\mathbf{x})$ 로 생각해보자.
- ▶ Majorizing 함수, $g(\mathbf{x}|\mathbf{x}_{(r)})$ 는 다음과 같은 성질을 만족하는 함수이다.
 - $\mathbf{x}_{(r)}$ 에서 f 와 같고 다른 \mathbf{x} 에서는 f 보다 큰 함수, 즉, (1) $g(\mathbf{x}_{(r)}|\mathbf{x}_{(r)}) = f(\mathbf{x}_{(r)})$, (2) $g(\mathbf{x}|\mathbf{x}_{(r)}) \geq f(\mathbf{x})$, for $\mathbf{x} \neq \mathbf{x}_{(r)}$.



- ▶ MM알고리즘은 $g(\mathbf{x}|\mathbf{x}_{(r)})$ 를 최소화하는 \mathbf{x} 값으로 업데이트한다. 즉,

$$\mathbf{x}_{(r+1)} = \arg \min_{\mathbf{x}} g(\mathbf{x}|\mathbf{x}_{(r)}).$$



- ▶ MM 알고리즘의 단조감소
- ▶ MM 알고리즘으로 업데이트 한 $\mathbf{x}_{(r+1)}$ 는 $f(\mathbf{x})$ 값을 감소시킨다.

$$f(\mathbf{x}_{(r)}) = g(\mathbf{x}_{(r)}|\mathbf{x}_{(r)}) \geq g(\mathbf{x}_{(r+1)}|\mathbf{x}_{(r)}) \geq f(\mathbf{x}_{(r+1)})$$

- ▶ 참고로 $\mathbf{x}_{(r+1)}$ 는 $g(\mathbf{x}|\mathbf{x}_{(r)})$ 를 최소화 시키지 않아도 $g(\mathbf{x}_{(r)}|\mathbf{x}_{(r)}) \geq g(\mathbf{x}_{(r+1)}|\mathbf{x}_{(r)})$ 를 만족하기만 하면 $f(\mathbf{x}_{(r)}) \geq f(\mathbf{x}_{(r+1)})$ 이 된다.

- ▶ $g(\mathbf{x}|\mathbf{x}_{(r)})$ 를 도입하는 이유
- (1) 실제 함수 ($f(\mathbf{x})$)가 계산이 어려운 경우, 최소화시키기 쉬운 $g(\mathbf{x}|\mathbf{x}_{(r)})$ 를 찾는다.
- (2) 볼록함수중에서 미분가능하지 않은 함수의 경우 미분가능한 $g(\mathbf{x}|\mathbf{x}_{(r)})$ 를 찾는다.
- ▶ 그럼 어떻게 $g(\mathbf{x}|\mathbf{x}_{(r)})$ 을 찾을까?
- ▶ 각 최적화 문제 마다 몇가지 방법들을 이용하여 찾는다.

MM 알고리즘의 예: LASSO 회귀분석

- ▶ LASSO (Least Absolute Shrinkage and Selection Operator) 회귀분석은 ℓ_1 페널티 함수를 도입하여 변수선택을 같이하는 방법이다.
- ▶ $y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i$ 에서 아래의 목적함수를 최소화 하는 $\boldsymbol{\beta}$ 를 추정한다.
- ▶ $S_\lambda(\boldsymbol{\beta}) = \sum_{i=1}^n (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^p |\beta_j|$

- ▶ Majorize 함수를 찾기위해 다음의 성질을 이용한다.

$$|\beta_j| \leq \frac{\beta_j^2}{2|\beta_{(r)j}|} + \frac{|\beta_{(r)j}|}{2}$$

- ▶ 이를 이용하면, 다음과 같은 미분가능한 majorize 함수를 찾을수 있다.

$$S_\lambda(\beta) \leq \sum_{i=1}^n (y_i - \mathbf{x}_i^T \beta)^2 + \lambda \frac{\beta_j^2}{2|\beta_{(r)j}|} + \frac{|\beta_{(r)j}|}{2} = g(\beta|\beta_{(r)})$$

EM 알고리즘

- ▶ EM (Expectation-Maximization) 알고리즘은 잠재변수 (latent variable)나 결측치 (missing variable)가 있는 경우의 가능도 함수 (likelihood function)를 최대화 할때 (MLE를 구할때) 사용하는 알고리즘이다.
- ▶ 다양한 통계분석방법 - 혼합분포 모형(mixture model), 군집분석 (cluster analysis), 임의효과 모형 (random effects model), 인과모형 (casual inference)등에서 사용된다.

혼합분포를 이용한 군집분석 예

EM알고리즘을 적용하는 예로 4장에서 다루었던 가우시안 혼합분포를 이용한 군집분석을 다시 소개한다.

- ▶ 두개의 군집으로 이루어져있는 데이터를 두 개의 구성원을 갖는 가우시안 혼합분포를 따른다고 가정하자.
- ▶ 즉, $X_1, \dots, X_n \stackrel{i.i.d.}{\sim} f(x|\theta)$, $f(x|\theta) = w_0 \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{1}{2\sigma_0^2}(X_i - \mu_0)^2} + (1 - w_0) \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{1}{2\sigma_1^2}(X_i - \mu_1)^2}$,
 $\theta = (\mu_0, \sigma_0^2, \mu_1, \sigma_1^2, w_0)$.

- ▶ 로그 가능도함수는 다음과 같다.

$$\begin{aligned}\ell(\theta) &= \sum_{i=1}^n \log f(X_i|\theta) \\ &= \sum_{i=1}^n \log \left(w_0 \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{1}{2\sigma_0^2}(X_i-\mu_0)^2} \right. \\ &\quad \left. + (1-w_0) \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{1}{2\sigma_1^2}(X_i-\mu_1)^2} \right)\end{aligned}$$

- ▶ 로그함수 내부에 복잡한 함수형태가 또 들어있어서 최대가능도를 최대가 되게 하는 θ 를 찾기가 쉽지 않다.

i -번째 데이터 X_i 가 어느 구성원으로부터 왔는지를 잠재 변수 (latent variable) K_i 를 도입하여 표현할 수 있다.

- ▶ $K_i = 0$ 이면 $X_i \sim N(\mu_0, \sigma_0^2)$
- ▶ $K_i = 1$ 이면 $X_i \sim N(\mu_1, \sigma_1^2)$
- ▶ 즉, K_i 는 i -번째 데이터가 어느 군집에 속하는지 알려주는 label로 생각할 수 있다.
- ▶ 잠재변수 K_i 는 $P(K_i = 0) = w_0, P(K_i = 1) = 1 - w_0$ 인 확률변수이다.
- ▶ EM알고리즘을 사용하여 $f(X_i|\theta)$ 보다 간단한 형태인 $f(X_i, K_i|\theta)$ 를 이용한 최대화 문제로 바꾸게 된다. (구체적 내용은 EM알고리즘 소개한후 다시 소개)

EM 알고리즘을 설명하기 위해 필요한 용어들

- ▶ \mathbf{Y}_o : 관측한 데이터 (앞의 예제- $\mathbf{Y}_o = (X_1, \dots, X_n)$)
- ▶ \mathbf{Y}_m : 결측치 또는 잠재변수 (앞의 예제- $\mathbf{Y}_m = (K_1, \dots, K_n)$)
- ▶ $\mathbf{Y}_c = (\mathbf{Y}_o, \mathbf{Y}_m)$: 관측값과 결측값을 다 고려한 데이터
- ▶ $f(\mathbf{Y}_c|\theta)$: \mathbf{Y}_c 의 확률밀도함수 (앞의 예제 -
 $f(\mathbf{X}, \mathbf{K}|\theta) = \prod_{i=1}^n f(X_i, K_i|\theta)$)
- ▶ $f(\mathbf{Y}_o|\theta)$: \mathbf{Y}_o 의 확률밀도함수. $f(\mathbf{Y}_o|\theta) = \int f(\mathbf{Y}_c|\theta) d\mathbf{Y}_m$
- ▶ $f(\mathbf{Y}_m|\mathbf{Y}_o, \theta)$: \mathbf{Y}_m 의 조건부 확률밀도함수.
 $f(\mathbf{Y}_m|\mathbf{Y}_o, \theta) = \frac{f(\mathbf{Y}_c|\theta)}{f(\mathbf{Y}_o|\theta)}$

EM알고리즘에서 minorize함수 찾기

- ▶ MLE는 관측한 데이터로 구한 가능도 함수(likelihood)를 최대화 하는 모수를 찾는것이다.
- ▶ 즉, $\arg \max_{\theta} \ell(\theta) = \arg \max_{\theta} \log f(\mathbf{Y}_o|\theta)$.
- ▶ $\ell(\theta) = f(\mathbf{Y}_o|\theta)$ 의 계산이 복잡한 겨우, $\ell(\theta) \geq Q(\theta|\theta_{(r)})$ 이고, $\ell(\theta_{(r)}) = Q(\theta_{(r)}|\theta_{(r)})$ 인 Q 를 찾아 Q 를 최대화하는 θ 로 업데이트 한다.
- ▶ 즉, MM알고리즘에서 minorize함수를 찾는것과 같다.

$$\begin{aligned}
\ell(\boldsymbol{\theta}) &= \log f(\mathbf{Y}_o|\boldsymbol{\theta}) = \log \int f(\mathbf{Y}_o, \mathbf{Y}_m|\boldsymbol{\theta}) d\mathbf{Y}_m \\
&= \log \int \frac{f(\mathbf{Y}_o, \mathbf{Y}_m|\boldsymbol{\theta})}{f(\mathbf{Y}_m|\mathbf{Y}_o, \boldsymbol{\theta}_{(r)})} f(\mathbf{Y}_m|\mathbf{Y}_o, \boldsymbol{\theta}_{(r)}) d\mathbf{Y}_m \\
&\geq \int \log \left(\frac{f(\mathbf{Y}_o, \mathbf{Y}_m|\boldsymbol{\theta})}{f(\mathbf{Y}_m|\mathbf{Y}_o, \boldsymbol{\theta}_{(r)})} \right) f(\mathbf{Y}_m|\mathbf{Y}_o, \boldsymbol{\theta}_{(r)}) d\mathbf{Y}_m \\
&\quad \text{by Jensen's inequality}
\end{aligned}$$

$$\begin{aligned}
\log \int \frac{f(\mathbf{Y}_o, \mathbf{Y}_m|\boldsymbol{\theta})}{f(\mathbf{Y}_m|\mathbf{Y}_o, \boldsymbol{\theta}_{(r)})} f(\mathbf{Y}_m|\mathbf{Y}_o, \boldsymbol{\theta}_{(r)}) d\mathbf{Y}_m &= \log E \left(\frac{f(\mathbf{Y}_o, \mathbf{Y}_m|\boldsymbol{\theta})}{f(\mathbf{Y}_m|\mathbf{Y}_o, \boldsymbol{\theta}_{(r)})} \right) \\
\int \log \left(\frac{f(\mathbf{Y}_o, \mathbf{Y}_m|\boldsymbol{\theta})}{f(\mathbf{Y}_m|\mathbf{Y}_o, \boldsymbol{\theta}_{(r)})} \right) f(\mathbf{Y}_m|\mathbf{Y}_o, \boldsymbol{\theta}_{(r)}) d\mathbf{Y}_m &= E \left(\log \left(\frac{f(\mathbf{Y}_o, \mathbf{Y}_m|\boldsymbol{\theta})}{f(\mathbf{Y}_m|\mathbf{Y}_o, \boldsymbol{\theta}_{(r)})} \right) \right) \\
\log E \left(\frac{f(\mathbf{Y}_o, \mathbf{Y}_m|\boldsymbol{\theta})}{f(\mathbf{Y}_m|\mathbf{Y}_o, \boldsymbol{\theta}_{(r)})} \right) &\geq E \left(\log \left(\frac{f(\mathbf{Y}_o, \mathbf{Y}_m|\boldsymbol{\theta})}{f(\mathbf{Y}_m|\mathbf{Y}_o, \boldsymbol{\theta}_{(r)})} \right) \right)
\end{aligned}$$

$$\ell(\boldsymbol{\theta}_{(r)}) = \log f(\mathbf{Y}_o | \boldsymbol{\theta}_{(r)}) = \int \log f(\mathbf{Y}_o | \boldsymbol{\theta}_{(r)}) f(\mathbf{Y}_m | \mathbf{Y}_o, \boldsymbol{\theta}_{(r)}) d\mathbf{Y}_m$$

$$\begin{aligned} & \ell(\boldsymbol{\theta}) - \ell(\boldsymbol{\theta}_{(r)}) \\ & \geq \int \log \left(\frac{f(\mathbf{Y}_o, \mathbf{Y}_m | \boldsymbol{\theta})}{f(\mathbf{Y}_m | \mathbf{Y}_o, \boldsymbol{\theta}_{(r)})} \right) f(\mathbf{Y}_m | \mathbf{Y}_o, \boldsymbol{\theta}_{(r)}) d\mathbf{Y}_m \\ & \quad - \int \log f(\mathbf{Y}_o | \boldsymbol{\theta}_{(r)}) f(\mathbf{Y}_m | \mathbf{Y}_o, \boldsymbol{\theta}_{(r)}) d\mathbf{Y}_m \\ & \geq \int \log \left(\frac{f(\mathbf{Y}_o, \mathbf{Y}_m | \boldsymbol{\theta})}{f(\mathbf{Y}_m | \mathbf{Y}_o, \boldsymbol{\theta}_{(r)}) f(\mathbf{Y}_o | \boldsymbol{\theta}_{(r)})} \right) f(\mathbf{Y}_m | \mathbf{Y}_o, \boldsymbol{\theta}_{(r)}) d\mathbf{Y}_m \\ & = \int \log \left(\frac{f(\mathbf{Y}_o, \mathbf{Y}_m | \boldsymbol{\theta})}{f(\mathbf{Y}_o, \mathbf{Y}_m | \boldsymbol{\theta}_{(r)})} \right) f(\mathbf{Y}_m | \mathbf{Y}_o, \boldsymbol{\theta}_{(r)}) d\mathbf{Y}_m \end{aligned}$$

$$\begin{aligned}
\ell(\boldsymbol{\theta}) &\geq \ell(\boldsymbol{\theta}_{(r)}) + \int \log \left(\frac{f(\mathbf{Y}_o, \mathbf{Y}_m | \boldsymbol{\theta})}{f(\mathbf{Y}_o, \mathbf{Y}_m | \boldsymbol{\theta}_{(r)})} \right) f(\mathbf{Y}_m | \mathbf{Y}_o, \boldsymbol{\theta}_{(r)}) d\mathbf{Y}_m \\
&=: Q_0(\boldsymbol{\theta} | \boldsymbol{\theta}_{(r)}) \\
\ell(\boldsymbol{\theta}_{(r)}) &= Q_0(\boldsymbol{\theta}_{(r)} | \boldsymbol{\theta}_{(r)})
\end{aligned}$$

한편 $Q_0(\boldsymbol{\theta} | \boldsymbol{\theta}_{(r)})$ 에서 $\boldsymbol{\theta}$ 에 관련된 부분은

$$\int \log(f(\mathbf{Y}_o, \mathbf{Y}_m | \boldsymbol{\theta})) f(\mathbf{Y}_m | \mathbf{Y}_o, \boldsymbol{\theta}_{(r)}) d\mathbf{Y}_m =: Q(\boldsymbol{\theta} | \boldsymbol{\theta}_{(r)})$$

따라서

$$\operatorname{argmax}_{\boldsymbol{\theta}} Q_0(\boldsymbol{\theta} | \boldsymbol{\theta}_{(r)}) = \operatorname{argmax}_{\boldsymbol{\theta}} Q(\boldsymbol{\theta} | \boldsymbol{\theta}_{(r)})$$

정리하면, $\ell(\theta) = \log f(\mathbf{Y}_o|\theta)$ 를 최대화하는 EM알고리즘은 반복알고리즘(iterative algorithm)으로 (r) 번째 스텝에서 $(r+1)$ 번째 스텝으로의 업데이트는 다음의 두 과정을 통해 진행된다.

(1) **Expectation step:** $Q(\theta|\theta_{(r)})$ 찾기

$$\begin{aligned} Q(\theta|\theta_{(r)}) &= \int \log(f(\mathbf{Y}_o, \mathbf{Y}_m|\theta)) f(\mathbf{Y}_m|\mathbf{Y}_o, \theta_{(r)}) d\mathbf{Y}_m \\ &= E(\log(f(\mathbf{Y}_o, \mathbf{Y}_m|\theta))) \end{aligned}$$

- $Q(\theta|\theta_{(r)})$ 이 기댓값 ($E()$)로 표현되기 때문에 Expectation step이라고 부름

(2) **Maximization step:** $\theta_{(r+1)} = \arg \max_{\theta} Q(\theta|\theta_{(r)})$

- ▶ 문제예따라 위 두 스텝을 정리하여 한번에 표시할수 있다.
(혼합분포를 이용한 군집분석의 예)

가우시안 혼합분포를 이용한 군집분석에서의 EM 알고리즘

- ▶ $X_1, \dots, X_n \stackrel{i.i.d.}{\sim} f(x|\theta)$, $f(x|\theta) = w_0 \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{1}{2\sigma_0^2}(X_i - \mu_0)^2} + (1 - w_0) \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{1}{2\sigma_1^2}(X_i - \mu_1)^2}$,
 $\theta = (\mu_0, \sigma_0^2, \mu_1, \sigma_1^2, w_0)$.
- ▶ $X_i | K_i = 0 \sim N(\mu_0, \sigma_0^2)$
- ▶ $X_i | K_i = 1 \sim N(\mu_1, \sigma_1^2)$
- ▶ K_i 는 i -번째 데이터가 어느 군집에 속하는지 알려주는 label
- ▶ $P(K_i = 0) = w_0$, $P(K_i = 1) = 1 - w_0$

- ▶ $\mathbf{Y}_o = \mathbf{X} = (X_1, \dots, X_n)$, $\mathbf{Y}_m = \mathbf{K} = (K_1, \dots, K_n)$
- ▶ $f(\mathbf{Y}_o, \mathbf{Y}_m | \theta) = f(\mathbf{X}, \mathbf{K} | \theta) = \prod_{i=1}^n f(X_i, K_i | \theta)$

$$\begin{aligned}
 \log f(\mathbf{Y}_o, \mathbf{Y}_m | \theta) &= \sum_{i=1}^n \log f(X_i, K_i | \theta) \\
 &= -\frac{n}{2} \log(2\pi) - \frac{1}{2} \sum_{i=1}^n \log(\sigma_{K_i}^2) - \sum_{i=1}^n \left(\frac{1}{2\sigma_{K_i}^2} (X_i - \mu_{K_i})^2 \right) \\
 &\quad + \left(n - \sum_{i=1}^n K_i \right) \log(w_0) + \left(\sum_{i=1}^n K_i \right) \log(1 - w_0)
 \end{aligned}$$

$$\begin{aligned}
Q(\theta|\theta_{(r)}) &= \int \log(f(\mathbf{Y}_o, \mathbf{Y}_m|\theta)) f(\mathbf{Y}_m|\mathbf{Y}_o, \theta_{(r)}) d\mathbf{Y}_m \\
&= \int \left(\sum_{i=1}^n \log f(X_i, K_i|\theta) \right) f(\mathbf{K}|\mathbf{X}, \theta_{(r)}) d\mathbf{K} \\
&= \sum_{i=1}^n \left(\log f(X_i, K_i = 0|\theta) P(K_i = 0|X_i, \theta_{(r)}) \right. \\
&\quad \left. + \log f(X_i, K_i = 1|\theta) P(K_i = 1|X_i, \theta_{(r)}) \right)
\end{aligned}$$

- ▶ $\frac{\partial Q}{\partial \mu_0} = 0, \frac{\partial Q}{\partial \mu_1} = 0, \frac{\partial Q}{\partial \sigma_0^2} = 0, \frac{\partial Q}{\partial \sigma_1^2} = 0, \frac{\partial Q}{\partial w_0} = 0$ 을 풀면 다음과 같은 업데이트식이 구해진다.

$$\mu_{j(r+1)} = \frac{\sum_{i=1}^n X_i \lambda_j(X_i, \theta_{(r)})}{\sum_{i=1}^n \lambda_j(X_i, \theta_{(r)})}, j = 0, 1.$$

$$\sigma_{j(r+1)}^2 = \frac{\sum_{i=1}^n (X_i - \mu_{j(r+1)})^2 \lambda_j(X_i, \theta_{(r)})}{\sum_{i=1}^n \lambda_j(X_i, \theta_{(r)})}, j = 0, 1$$

$$w_{0(r+1)} = \frac{1}{n} \sum_{i=1}^n \lambda_0(X_i, \theta_{(r)}),$$

$$\lambda_j(X_i, \theta_{(r)}) = P(K_i = j | X_i, \theta_{(r)}), j = 0, 1.$$

몬테카를로 방법 (Monte Carlo method)

- ▶ 어떤 값 (또는 함수값)을 근사적으로 계산하는데 있어 확률분포로부터 생성한 무작위 샘플(표본)들을 이용하는 방법이다.
- ▶ 이를 위해 큰수의 법칙 (Law of large number, LLN)를 이용한다.
- ▶ LLN: $X_1, \dots, X_n \stackrel{i.i.d.}{\sim} f$ 에 대하여

$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i \xrightarrow{p} E(X_1)$$

- ▶ 구하고자 하는 값을 θ 라고 하자.
- ▶ θ 는 어떤 확률변수의 기대값으로 표현할수 있다고 하자. 즉, $\theta = E(h(X)), X \sim f$.
- ▶ 만약 X_1, \dots, X_n 을 f 로 부터 생성한 무작위 샘플(표본)이라면, LLN에 의해 다음이 성립한다.

$$\frac{1}{n} \sum_{i=1}^n h(X_i) \xrightarrow{P} E(h(X)) = \theta$$

- ▶ 따라서 몬테카를로 방법을 쓰기 위해서는 주어진 확률분포로부터 무작위 샘플을 생성하는 (난수생성) 알고리즘이 필요하다.
- ▶ 대부분의 알려진 확률분포들은 무작위 샘플을 생성하는 알고리즘들이 연구되어, R, Python등에서 쉽게 사용할수 있다.

Inverse CDF 방법

- ▶ Probability integral transform이론: $F_X(X) \sim U$, U 는 $(0, 1)$ 에서의 균일분포 ($Uniform(0, 1)$).
 $X \sim F_X^{-1}(U) = \inf\{x : F(x) \geq U\}$

- ▶ Inverse CDF 방법

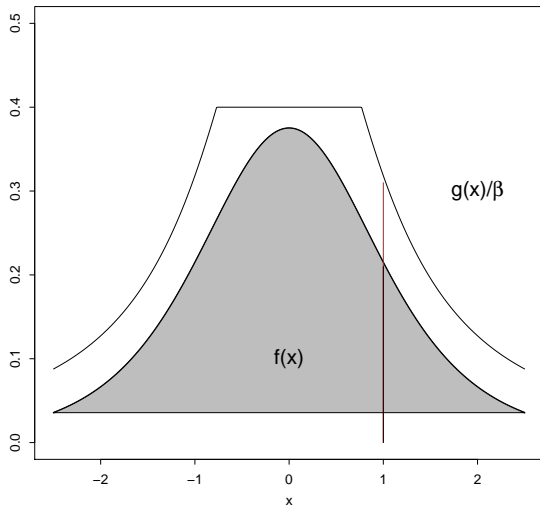
- (1) $f(x)$ 의 support내의 숫자열 x_1, \dots, x_m 을 적당히 찾아 $u_i = F(x_i)$ 를 구한다.
- (2) $U \sim Uniform(0, 1)$ 인 U 를 생성하여 $u_i \leq U \leq u_j$ 를 만족하는 가장 가까운 u_i, u_j 에 대하여 다음의 X 를 구한다.

$$X = \frac{u_j - U}{u_j - u_i} x_i + \frac{U - u_i}{u_j - u_i} x_j$$

- ▶ $(x_i, F(x_i)), (x_j, F(x_j))$ 를 linear interpolation 한 값이다.

Rejection sampling

- ▶ 확률분포 f 를 따르는 X 를 생성하고자 한다. $X \sim f$
- ▶ f 보다 난수생성이 쉽고 다음을 만족하는 확률분포 g 를 찾자.
 - $\beta f(x) \leq g(x), 0 < \beta < 1.$
- ▶ $g(x)/\beta = e(x)$ 를 $f(x)$ 의 upper envelop이라고 한다.



Rejection sampling 알고리즘

- (1) g 로부터 X 생성한다.
- (2) $Uniform(0, 1)$ 으로부터 U 를 생성한다.
- (3) 만약 $Ug(X) \leq \beta f(X)$ (또는 $U \leq f(X)/e(X)$)를 만족하면 X 를 f 로부터 생성된 난수로 받아들인다(accept).
아니라면(reject), (1)으로 돌아간다.

X 가 f 로부터의 난수인지 확인해보자.

▶ $P(\text{accept } X)$ 인 경우

$$P(\text{accept } X) = P(U \leq \beta f(X)/g(X)) = \int (\beta f(X)/g(X))g(x)dx = \beta \text{ 이므로}$$

$$\begin{aligned} P(X \leq x | \text{accept}) &= \frac{P(X \leq x \text{ and } U \leq \beta f(X)/g(X))}{P(\text{accept})} \\ &= \int_{-\infty}^x \frac{(\beta f(w)/g(w))g(w)}{\beta} dw \\ &= \int_{-\infty}^x f(w)dw = F(x) \end{aligned}$$

Importance sampling

- ▶ $\theta = E(h(X))$ 인 θ 를 구하는 문제로 다시 돌아가보자. 여기서 $X \sim f$ 로 가정한다.
- ▶ f 보다 난수생성이 쉽고 support가 f 의 support를 포함하는 확률분포 g 가 있다고 하자.
- ▶ θ 를 g 를 이용하여 다음과 같이 표현할 수 있다.

$$\begin{aligned}\theta &= E_f(h(X)) = \int h(x)f(x)dx \\ &= \int h(x)\frac{f(x)}{g(x)}g(x)dx = E_g\left(h(X)\frac{f(X)}{g(X)}\right)\end{aligned}$$

- ▶ X_1, \dots, X_n 이 g 로부터 생성된 난수라고 하자.
 $w^*(x) = f(x)/g(x)$ 라고 하면,

$$\hat{\theta} = \frac{1}{n} \sum_i h(X_i) w^*(X_i),$$

는 θ 를 근사(approximate)하는 값이 된다.

- ▶ g 를 importance sampling 함수라고 부른다.

변분 추론 (Variational inference)

- ▶ 베이지안분석은 데이터의 분포에 관한 모형 (Likelihood, $L(\theta|X)$)과 데이터에 관한 모형을 정하는 모수 (parameter)의 사전분포 (prior distribution, $\pi(\theta)$)를 이용하여 모수의 사후확률 (Posterior probability, $\pi(\theta|X)$)을 구하고 이를 이용하여 모수에 관한 추론을 하는 방법이다.
- ▶ 이때, Likelihood가 복잡하거나, 데이터가 아주 큰 경우, 일반적으로 사후확률을 (이론적/계산적) 구하기 어렵다.
- ▶ Markov Chain Monte Carlo (MCMC)방법은 근사 베이지안 방법(Approximate Bayesian method)의 일종으로 사후분포를 이론적으로 구하는 대신 마코프 체인을 만들어 생성한 샘플이 사후분포의 샘플이 되도록 하여 샘플을 이용하여 추론하는방법이다.
- ▶ 사후분포의 형태에 따라 깁스 샘플링 (Gibbs sampling) 또는 메트로폴리스-헤이스팅 샘플링 (Metropolis-Hastings sampling)등이 있다.

- ▶ MCMC방법은 복잡한 모형인 경우 또는 데이터가 큰 경우 계산속도가 느리다.
- ▶ 변분추론은 또다른 근사 베이지안 방법(Approximate Bayesian method)으로 사후분포에 가까운, 샘플링이 쉬운 분포를 찾아 추론을 하는것이며, 일반적으로 MCMC보다 속도가 빠르다.
- ▶ 베이지안분석에서 시작되었지만, 베이지안이 아니더라도 MCMC의 경우 분포로부터 표본을 샘플링하는 기법으로, 변분추론의 경우 분포를 근사시키는 기법으로 사용될수 있다.
- ▶ 본 강의는 "Variational inference: A review for statisticals" by Blei et al. (2017)을 참고하였음.

사후분포가 복잡한 모형인 경우의 예

- ▶ 가우시안 혼합분포 모형

데이터모형: $X_i \sim \pi_1 N(\mu_1, \sigma^2) + \cdots, + \pi_K N(\mu_K, \sigma^2),$
 $i = 1, \cdots, n.$

사전분포: $\mu_i \sim N(0, \tau^2)$

- ▶ 혼합분포의 어느 구성원으로부터 왔는지를 나타내는 Z_i (label)을 도입하면 가우시안 혼합분포 모형은 다음과 같이 계층적으로 표현할수 있다.

$$X_i | Z_i \sim N(\mu_{Z_i}, \sigma^2)$$

$$Z_i \sim \text{multi}(\pi), \pi = (\pi_1, \cdots, \pi_K)$$

$$\mu_i \sim N(0, \tau^2)$$

- ▶ 위 예제의 관심 사후분포는 $\boldsymbol{\mu} = (\mu_1, \dots, \mu_K)$, $\mathbf{Z} = (Z_1, \dots, Z_n)$ 의 사후분포이다.
- ▶ σ^2, τ^2 는 고정되어 있다고 가정하는 경우 (편의상)

$$p(\boldsymbol{\mu}, \mathbf{Z} | \mathbf{X}) = \frac{p(\mathbf{X} | \mathbf{Z}, \boldsymbol{\mu}) p(\mathbf{Z}) p(\boldsymbol{\mu})}{\int_{\boldsymbol{\mu}} \sum_{\mathbf{Z}} p(\mathbf{X} | \mathbf{Z}, \boldsymbol{\mu}) p(\mathbf{Z}) p(\boldsymbol{\mu}) d\boldsymbol{\mu}}$$

- ▶ 분자: $\prod_{i=1}^n p(X_i | Z_i) p(Z_i) \prod_{k=1}^K p(\mu_k)$
- ▶ 분모:

$$\begin{aligned} & \int_{\mu_K} \cdots \int_{\mu_1} \sum_{Z_1=1}^K \cdots \sum_{Z_n=1}^K \prod_{i=1}^n p(X_i | Z_i) p(Z_i) \\ & \quad \times \prod_{k=1}^K p(\mu_k) d\mu_1 \cdots d\mu_K \end{aligned}$$

- ▶ 분모에서 n^K 합 계산은 데이터가 아주 큰 경우 오래 걸린다.

모형 가정

- ▶ 변분추론 설명을 위해서 다음과 같은 가정을 한다.
 - 데이터: $\mathbf{X} = (X_1, \dots, X_n)$
 - 은닉변수(잠재변수): $\mathbf{Z} = (Z_1, \dots, Z_m)$
 - 추가 모수: α
 - 목적: 사후분포 $p(\mathbf{Z}|\mathbf{X}, \alpha)$ 와 가까운 다루기 쉬운 분포를 찾아서 필요에 따라 근사시킨 분포를 이용하여 \mathbf{Z} 를 생성하거나 사후분포의 특성값들을 근사적으로 구한다.

근사분포 찾기

- ▶ $p(\mathbf{Z}|\mathbf{X}, \alpha)$ 와 가까운 $q(\mathbf{Z}|\nu)$ 를 찾는다고 하자.
- ▶ 이때, $q(\mathbf{Z}|\nu)$ 를 ν 에 따라 움직이는 분포들의 모임 (클래스 \mathcal{Q} 라고 하자)이라고 보면, 이러한 분포들 중 $p(\mathbf{Z}|\mathbf{X}, \alpha)$ 에 가장 가깝도록 하는 ν 를 찾는 문제로 볼수도 있다.
- ▶ 여기서 ν 는 변분 모수 (variational parameter)라 부른다.
- ▶ 두 분포가 가깝다는 기준, 즉 분포사이의 "가까움"을 나타내는 기준이 필요하다. 이를 위해 KL divergence를 소개한다.

쿨백-라이블러 발산 (Kullback-Leibler Divergence)

- ▶ 정보이론 (Information theory)에서 온 개념
- ▶ 두 분포사이의 "가까움"을 나타내는 값

$$\begin{aligned} KL(q \parallel p) &= E_q \left(\log \frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{X})} \right) \\ &= E_q (\log q(\mathbf{Z}) - \log p(\mathbf{Z}|\mathbf{X})) \\ &= \sum_{\mathbf{z}} \log \left(\frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{X})} \right) q(\mathbf{z}) \quad (\text{discrete}) \\ &= \int \log \left(\frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{X})} \right) q(\mathbf{z}) d\mathbf{z} \quad (\text{continuous}) \end{aligned}$$

- ▶ 만약 $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X})$ 라면 $KL(q \parallel p) = 0$.
- ▶ $KL(q \parallel p) \geq 0$

$$\begin{aligned}
 KL(q \parallel p) &= E_q \left(\log \frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{X})} \right) \\
 &= E_q \left(-\log \frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} \right) \\
 &\geq -\log E_q \left(\frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} \right) \quad (\text{Jensen's inequality}) \\
 &= -\log \int \frac{p(\mathbf{z}|\mathbf{X})}{q(\mathbf{z})} q(\mathbf{z}) d\mathbf{z} \\
 &= -\log \int p(\mathbf{z}|\mathbf{X}) d\mathbf{z} = -\log(1) = 0.
 \end{aligned}$$

- ▶ $KL(q \parallel p) \neq KL(p \parallel q)$. 비대칭이므로 "거리(distance)"로 볼 수 없다.

Evidence Lower Bound (ELBO)

- ▶ $q^* = \operatorname{argmin}_{q \in \mathcal{Q}} KL(q(\mathbf{Z}) \parallel p(\mathbf{Z}|\mathbf{X}))$
- ▶ $KL(q(\mathbf{Z}) \parallel p(\mathbf{Z}|\mathbf{X}))$ 계산을 위해서는 $\log(p(\mathbf{Z}|\mathbf{X})) = \log(p(\mathbf{Z}, \mathbf{X})/p(\mathbf{X}))$ 를 계산해야 한다.
- ▶ $\log(p(\mathbf{X}))$ 의 계산이 일반적으로 복잡하다

- ▶ 한편 $KL(q(\mathbf{Z}) \parallel p(\mathbf{Z}|\mathbf{X})) \geq 0$ 으로부터

$$\begin{aligned} KL(q(\mathbf{Z}) \parallel p(\mathbf{Z}|\mathbf{X})) &= E_q \left(\log \frac{q(\mathbf{Z})}{p(\mathbf{Z}|\mathbf{X})} \right) \\ &= E_q(\log q(\mathbf{Z})) - E_q(\log p(\mathbf{Z}|\mathbf{X})) \\ &= E_q(\log q(\mathbf{Z})) - E_q(\log p(\mathbf{Z}, \mathbf{X})) + E_q(\log p(\mathbf{X})) \\ &= E_q(\log q(\mathbf{Z})) - E_q(\log p(\mathbf{Z}, \mathbf{X})) + \log p(\mathbf{X}) \\ &= \log p(\mathbf{X}) - (E_q(\log p(\mathbf{Z}, \mathbf{X})) - E_q(\log q(\mathbf{Z}))) \geq 0 \end{aligned}$$

- ▶ $\log p(\mathbf{X}) \geq E_q(\log p(\mathbf{Z}, \mathbf{X})) - E_q(\log q(\mathbf{Z}))$
- ▶ $\log p(\mathbf{X})$ 는 관측값의 Likelihood로 evidence라고도 부름.
- ▶ 따라서, $E_q(\log p(\mathbf{Z}, \mathbf{X})) - E_q(\log q(\mathbf{Z}))$ 는 evidence의 lower bound가 된다.

- ▶ $ELBO(q) = E_q(\log p(\mathbf{Z}, \mathbf{X})) - E_q(\log q(\mathbf{Z}))$
- ▶ $KL(q(\mathbf{Z}) \parallel p(\mathbf{Z}|\mathbf{X})) = -ELBO(q) + \log p(\mathbf{X})$ 이 된다.
- ▶ $\log p(\mathbf{X})$ 는 q 에 대해서 상수이므로 KL 을 최소화 시키는 q 를 찾는데 필요가 없다.
- ▶ 따라서

$$q^* = \operatorname{argmin}_{q \in \mathcal{Q}} KL(q(\mathbf{Z}) \parallel p(\mathbf{Z}|\mathbf{X})) = \operatorname{argmax}_{q \in \mathcal{Q}} ELBO(q(\mathbf{Z}))$$

ELBO의 이해

$$\begin{aligned} ELBO(q) &= E_q(\log p(\mathbf{Z}, \mathbf{X})) - E_q(\log q(\mathbf{Z})) \\ &= E_q(\log p(\mathbf{X}|\mathbf{Z})) + E_q(\log p(\mathbf{Z})) - E_q(\log q(\mathbf{Z})) \\ &= E_q(\log p(\mathbf{X}|\mathbf{Z})) - E_q(\log(q(\mathbf{Z})/p(\mathbf{Z}))) \\ &= E_q(\log p(\mathbf{X}|\mathbf{Z})) - KL(q(\mathbf{Z}) \parallel p(\mathbf{Z})) \end{aligned}$$

- ▶ 첫번째 항은 관측값 \mathbf{X} 의 잠재변수 \mathbf{Z} 가 주어졌을때의 Likelihood의 기대값으로 볼수 있다.
- ▶ 따라서 \mathbf{Z} 값이 $p(\mathbf{X}|\mathbf{Z})$ 를 크게 하도록 하 (Likelihood를 증가시키는 또는 데이터 \mathbf{X} 를 더 잘 설명하는)는 $q(\mathbf{Z})$ 를 찾으려 한다.
- ▶ 두번째 항은 \mathbf{Z} 의 사전분포 $p(\mathbf{Z})$ 와 $q(\mathbf{Z})$ 사이의 KL 발산이다.
- ▶ 따라서, 사전분포 $p(\mathbf{Z})$ 와 가까운 $q(\mathbf{Z})$ 를 찾으려 한다.
- ▶ $ELBO(q)$ 는 likelihood와 prior사이에서 적절한 q 를 찾게된다.

ELBO 최대화를 위한 \mathcal{Q}

$$q^* = \operatorname{argmin}_{q \in \mathcal{Q}} KL(q(\mathbf{Z}) \parallel p(\mathbf{Z}|\mathbf{X})) = \operatorname{argmax}_{q \in \mathcal{Q}} ELBO(q(\mathbf{Z}))$$

- ▶ ELBO 최대화 문제에서 $q \in \mathcal{Q}$ 를 찾는것은 \mathcal{Q} 를 어떻게 분포 집합 (probability distribution family)으로 놓느냐에 따라 계산 복잡도가 달라진다.
- ▶ 본 강의에서는 mean-field variational family인 경우를 소개한다.
- ▶ mean-field variational family는 잠재변수가 서로 독립이면서 각기 다른 변분인자 (variational factor)에 의존하는것으로 가정한다. 즉, $q(\mathbf{Z}) = \prod_{j=1}^m q_j(Z_j)$.
- ▶ 따라서 $ELBO(q)$ 를 최대화하는 $\{q_j^*\}$ 를 찾는 문제로 생각한다.

- ▶ variational family는 데이터 \mathbf{X} 에 의존하지 않는다.
- ▶ mean-field variational family보다 복잡한 family를 고려할수도 있으나 계산상의 복잡도가 커진다.
- ▶ 구체적으로 어떤 variational family (확률분포)를 고려할지는 문제에 따라 다르다.
- ▶ variational family가 정해지면 최대화 시키는 최적화 알고리즘을 상황에 맞게 적용한다.
- ▶ 주어진 데이터 \mathbf{X} 와, variational family Q 가 정해져서 $\{q_j^*\}$ 를 찾으면 필요에 따라 $\{q_j^*\}$ 를 이용하여 Z_i 를 생성할수 있다.
- ▶ 생성된 Z_i 를 이용하여 데이터 \mathbf{X} 와 유사한 \mathbf{X}^* 를 생성할수도 있다 (generative model).

ELBO 최대화 알고리즘

- ▶ Q 가 mean-field variational family인 경우 ELBO를 최대화하는 알고리즘으로 coordinate ascent 알고리즘을 소개한다.
- ▶ 반복알고리즘으로 $q_j(Z_j)$, $j \neq i$ 가 주어졌을때 $q_i(Z_i)$ 를 찾는것을 $i = 1, \dots, m$ 로 반복한다.
- ▶ 이를 위해 다음의 두 식을 이용한다.

$$p(\mathbf{Z}, \mathbf{X}) = \left(\prod_{j=1}^m p(Z_j | \mathbf{Z}_1^{j-1}, \mathbf{X}) \right) p(\mathbf{X})$$

$$q(\mathbf{Z}) = \prod_{j=1}^m q_j(Z_j)$$

$$\begin{aligned}
ELBO(q) &= E_q(\log p(\mathbf{Z}, \mathbf{X})) - E_q(\log q(\mathbf{Z})) \\
&= E_q \log p(\mathbf{X}) + \sum_{j=1}^m E_q \log p(Z_j | \mathbf{Z}_1^{j-1}, \mathbf{X}) - \sum_{j=1}^m E_{q_j} \log q_j(Z_j) \\
&= E_q \log p(\mathbf{X}) + \sum_{j=1}^m \int q_j(z_j) \left(\log p(Z_j | \mathbf{Z}_1^{j-1}, \mathbf{X}) \right) \\
&\quad \times \prod_{i \neq j} q_i(z_i) dz_i dz_j - \sum_{j=1}^m E_{q_j} \log q_j(Z_j) \\
&= E_q \log p(\mathbf{X}) + \sum_{j=1}^m \int q_j(z_j) E_{-j}(\log p(Z_j | \mathbf{Z}_{-j}, \mathbf{X})) dz_j \\
&\quad - \sum_{j=1}^m E_{q_j} \log q_j(Z_j)
\end{aligned}$$

- ▶ $\int q_i(z_i) dz_i = 1$ 인 조건하에서 $q_j(Z_j), j \neq i$ 가 주어졌을때 $ELBO(q)$ 를 최대화 시키는 $\{q_i(z_i)\}$ 를 찾으면 다음과 같다.

$$\begin{aligned} q_i^*(z_i) &\propto \exp(E_{-i}(\log p(Z_i | \mathbf{Z}_{-i}, \mathbf{X}))) \\ &\propto \exp(E_{-i}(\log p(Z_i, \mathbf{Z}_{-i}, \mathbf{X}))) \end{aligned}$$

- ▶ $q_i^{(r)}(z_i), i = 1, \dots, m$ 을 반복적으로 찾으면서 $EBLO(q^{(r)})$ 을 계산해서 수렴할때까지 (r) 을 증가시키면서 업데이트 한다.