Module 12

Latent Variabl Model

Kentaro Nakamura

GOV 2003

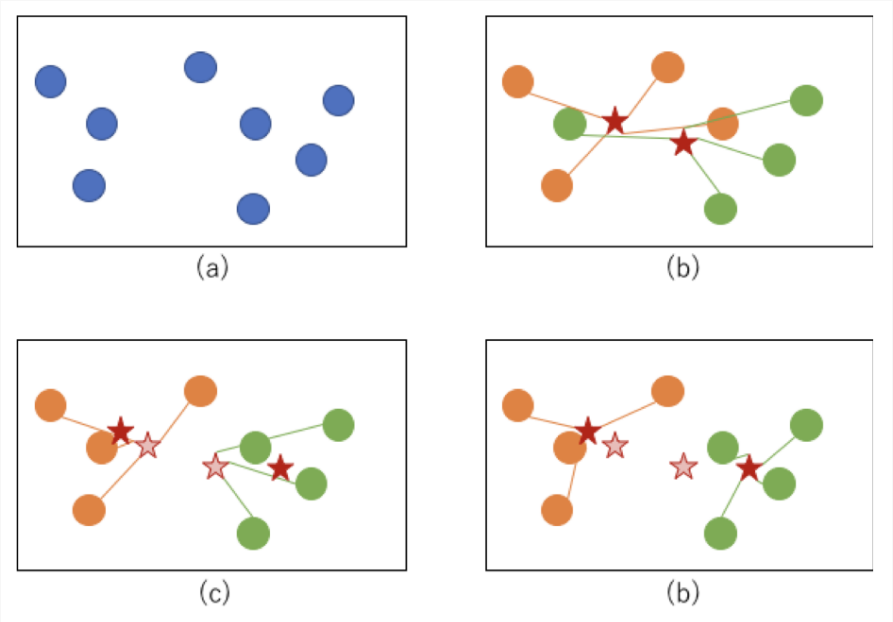
May 4th, 2026

Agenda

- k-means algorithm
- Gaussian Mixture Models (GMM)
- EM Algorithm in general
  - How to derive E / M-step
  - Why EM algorithm works
  - Uncertainty quantification under EM
- Useful resource: Bishop (2006) Chapter 9
- For future reference (not for final exam, but in appendix)
  - Variational Inference
  - MCMC Methods

k-means algorithm (Overview)

- **Goal**: discover the latent cluster that separates the data well
  - Input: $\boldsymbol{X}_i$ (observed data), K (number of clusters)
- **Assumption**: Each observation belongs to one cluster (**hard clustering**)
- **Algorithm** (**Lloyd's algorithm**)
  - Randomly assign each observation to one of K clusters.
  - Compute the centroid of the observations assigned to each cluster.
  - Reassign each observation to the cluster with the nearest centroid.
  - Repeat steps 2 and 3 until the cluster assignments no longer change.



k-means algorithm: optimization (1)

- Formally, the objective function is written as

$$\{\boldsymbol{Z}, \boldsymbol{\mu}\} := \arg\min_{\boldsymbol{Z},\boldsymbol{\mu}} \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbf{1}\{Z_i = k\} ||\boldsymbol{X}_i - \boldsymbol{\mu}_k||_2^2$$

  - Let's see why Lloyd's algorithm solves this objective function.

- Now, let's fix the assignment $\boldsymbol{Z}$. Then, the derivative w.r.t. μ_k is

$$\frac{\partial}{\partial \mu_k} \sum_{i:Z_i=k} ||\boldsymbol{X}_i - \boldsymbol{\mu}_k||_2^2 = -2 \cdot \sum_{i:Z_i=k} (\boldsymbol{X}_i - \mu_k)$$

  which means that setting $\mu_k = \frac{1}{n_k} \sum_{i:Z_i=k} \boldsymbol{X}_i$ (equivalent to STEP 2 in Lloyd's algorithm) achieves the first order condition

k-means algorithm: optimization (2)

- Now, think about optimizing cluster assignment given the centroid

$$\{\boldsymbol{Z}^{(t+1)}\} := \arg\min_{\boldsymbol{Z}} \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbf{1}\{Z_i = k\} ||\boldsymbol{X}_i - \boldsymbol{\mu}_k^{(t)}||_2^2$$

- Note that the objective function is separable across i:

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbf{1}\{Z_i = k\} ||\boldsymbol{X}_i - \boldsymbol{\mu}_k^{(t)}||_2^2 = \sum_{i=1}^{n} \min_{k \in \{1,\ldots,K\}} ||\boldsymbol{X}_i - \boldsymbol{\mu}_k^{(t)}||_2^2$$

- Therefore, for each observation i, we can optimize independently:

$$Z_i^{(t+1)} = \arg \min_{k \in \{1,\ldots,K\}} ||\boldsymbol{X}_i - \boldsymbol{\mu}_k^{(t)}||_2^2$$

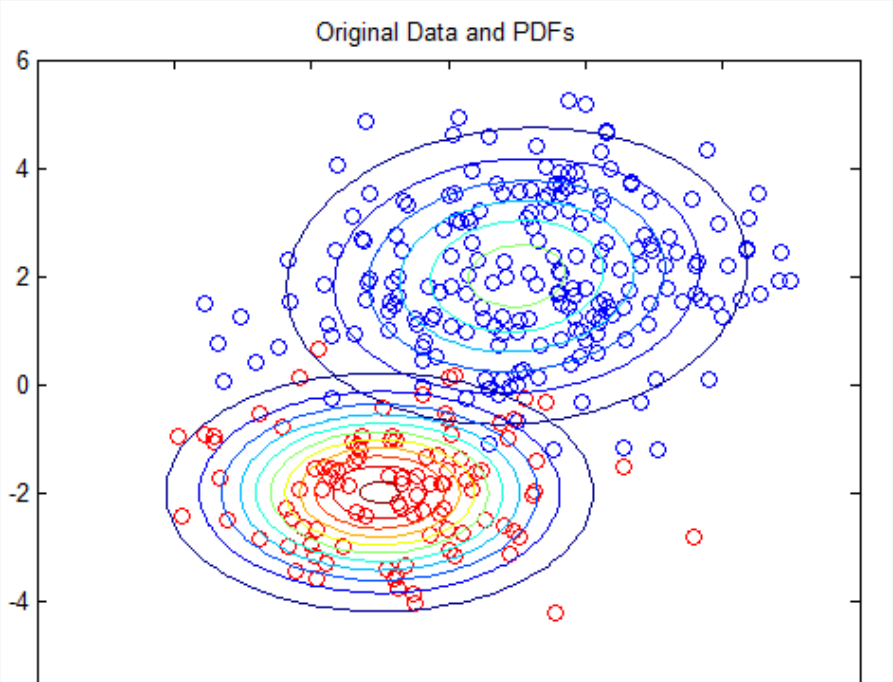
- This is exactly STEP 3 in Lloyd's algorithm (assign to nearest centroid)
  - Note that this might yield local optimal depending on initialization, so typically repeat the procedure with multiple initialization and pick up the one that minimizes the objective function the most.

Gaussian (Finite) Mixture Models

- We want to extend k-means by allowing each observations to belong to multiple cluster
- **Gaussian Mixture Models** (GMM): The entire data generating process is

$$\boldsymbol{X}_i \mid Z_i, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K \sim \mathcal{N}(\boldsymbol{\mu}_{Z_i}, \boldsymbol{\Sigma}_{Z_i})$$
$$Z_i \sim \text{Categorical}(\boldsymbol{\pi})$$

where $\boldsymbol{\pi} = [\pi_1, \cdots, \pi_K]^\top$, $\sum_{k=1}^K \pi_k = 1$ and $0 \leq \pi_k \leq 1$.



Likelihood of Latent Variable Model (1)

- Let's start with the likelihood of GMM as if we observe latent variable.

$$\mathcal{L}^{(\text{full})}(\boldsymbol{X}, \boldsymbol{Z}; \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{i=1}^{n} \prod_{k=1}^{K} \left(\pi_k \mathcal{N}(X_i; \mu_k, \Sigma_k)\right)^{\mathbf{1}\{Z_i = k\}}$$

- Thus, the log-likelihood is given by

$$\ell^{(\text{full})}(\boldsymbol{X}, \boldsymbol{Z}; \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbf{1}\{Z_i = k\} \left(\log \pi_k + \log \mathcal{N}(X_i; \mu_k, \Sigma_k)\right)$$

- Thus, if we observe Z_i, then we get MLE

$$\hat{\pi}_k = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{Z_i = k\}, \qquad \hat{\mu}_k = \frac{\sum_{i=1}^{n} \mathbf{1}\{Z_i = k\} X_i}{\sum_{i=1}^{n} \mathbf{1}\{Z_i = k\}}$$

$$\hat{\Sigma}_k = \frac{1}{\sum_{i=1}^{n} \mathbf{1}\{Z_i = k\}} \sum_{i=1}^{n} \mathbf{1}\{Z_i = k\} (X_i - \hat{\mu}_k)(X_i - \hat{\mu}_k)^\top$$

  - So, if we observe Z_i, then the problem is easy

Likelihood of Latent Variable Model (2)

- However, Z_i is latent. We cannot observe. What we observe is the observed likelihood:

$$\begin{aligned}
\mathcal{L}^{(\text{obs})}(\boldsymbol{X}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) &= \sum_{\boldsymbol{z}} \mathcal{L}^{(\text{full})}(\boldsymbol{X}, \boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) \\
&= \sum_{z_1, \cdots, z_n} \prod_{i=1}^{n} \prod_{k=1}^{K} \left(\pi_k \mathcal{N}(X_i; \mu_k, \Sigma_k)\right)^{\mathbf{1}\{z_i = k\}} \\
&= \sum_{z_1, \cdots, z_n} \prod_{i=1}^{n} \left(\pi_{z_i} \mathcal{N}(X_i; \mu_{z_i}, \Sigma_{z_i})\right) \\
&= \prod_{i=1}^{n} \sum_{z_i} \left(\pi_{z_i} \mathcal{N}(X_i; \mu_{z_i}, \Sigma_{z_i})\right) \\
&= \prod_{i=1}^{n} \sum_{k} \pi_k \mathcal{N}(X_i; \mu_k, \Sigma_k)
\end{aligned}$$

where the last equality is by re-indexing.

  - Recall that $\Pr(X = x) = \sum_y \Pr(X_i = x, Y_i = y)$ (this is why you need to integrate out the latent)

Problem in Latent Variable Model

- Therefore, the observed data log likelihood is given by

$$\log p(\boldsymbol{X} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{i=1}^{n} \log \left\{ \sum_{k} \pi_k \mathcal{N}(X_i; \mu_k, \Sigma_k) \right\}$$

- This is hard to maximize because of the structure log (sum)
  - The log of sum over convex function is generally not convex
  - To see why, consider the example

$$f(x) = \log\{\exp(-[x-2]^2) + \exp(-([x+2]^2)\}$$

    where you see each term is convex
  - When $x = 2 \rightarrow$ first term dominates (peak)
  - When $x = -2 \rightarrow$ second term dominates (peak)

- Similarly, GMM suffers from non-convexity problem
  - There are K latent cluster, but each cluster label does not have meaning
  - **Label switching**: There are $K!$ equivalent solutions

EM Algorithm for GMM (1)

- **Goal**: Maximize the observed data likelihood $\ln p(\boldsymbol{X} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})$
- Now, recall that we have

$$\ln p(\boldsymbol{X} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{i=1}^{n} \log \left\{ \sum_{k} \pi_k \mathcal{N}(\boldsymbol{X}_i; \mu_k, \Sigma_k) \right\}$$

- Now, check the first order condition for each parameter. The derivative w.r.t. $\boldsymbol{\mu}_k$ is given by

$$0 = -\sum_{i=1}^{n} \underbrace{\frac{\pi_k \mathcal{N}(\boldsymbol{X}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{X}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{:=\gamma_{i,k}(=\Pr(Z_i=k|\boldsymbol{X}_i))} \boldsymbol{\Sigma}_k^{-1}(X_i - \boldsymbol{\mu}_k)$$

  where $\gamma_{i,k}$ is called responsibility parameter.

- Similarly, the derivative with respect to $\boldsymbol{\mu}_k$ yields

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{n} \gamma_{i,k} \boldsymbol{X}_i}{\sum_{i=1}^{n} \gamma_{i,k}}$$

EM Algorithm for GMM (2)

- Next, let's derive the MLE for $\boldsymbol{\Sigma}_k$.
- The first order condition with respect to $\boldsymbol{\Sigma}_k^{-1}$ is

$$\begin{aligned} 0 &= \sum_{i=1}^{n} \gamma_{i,k} \frac{\partial}{\partial \boldsymbol{\Sigma}_k^{-1}} \log \mathcal{N}(\boldsymbol{X}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ &= \frac{1}{2} \sum_{i=1}^{n} \gamma_{i,k} \left[\boldsymbol{\Sigma}_k - (\boldsymbol{X}_i - \boldsymbol{\mu}_k)(\boldsymbol{X}_i - \boldsymbol{\mu}_k)^\top \right]. \end{aligned}$$

- Therefore,

$$\sum_{i=1}^{n} \gamma_{i,k} \boldsymbol{\Sigma}_k = \sum_{i=1}^{n} \gamma_{i,k} (\boldsymbol{X}_i - \boldsymbol{\mu}_k)(\boldsymbol{X}_i - \boldsymbol{\mu}_k)^\top.$$

- This gives

$$\boldsymbol{\Sigma}_k = \frac{\sum_{i=1}^{n} \gamma_{i,k} (\boldsymbol{X}_i - \boldsymbol{\mu}_k)(\boldsymbol{X}_i - \boldsymbol{\mu}_k)^\top}{\sum_{i=1}^{n} \gamma_{i,k}}.$$

EM Algorithm for GMM (3)

- For the membership probability π_k, we need to incorporate constraints.
  - Using Lagrange multiplier and maximizing

$$\log p(\boldsymbol{X} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) + \lambda(\sum_{k=1}^{K} - 1)$$

    which yields

$$0 = \sum_{i=1}^{n} \frac{\mathcal{N}(\boldsymbol{X}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{X}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} + \lambda$$

- Now, multiplying both sides by π_k, we get

$$0 = \sum_{i=1}^{n} \gamma_{i,k} + \lambda \pi_k$$

- Since $\sum \pi_k = 1$, $\sum_k \{\sum_{i=1}^{n} \gamma_{i,k} + \lambda \pi_k\} = \sum_k \sum_{i=1}^{n} \gamma_{i,k} + \lambda = 0$.
  - I.e., $\lambda = -N$
  - This gives us $\pi_k = \sum_{i=1}^{n} \gamma_{i,k} / N$.

EM Algorithm for GMM (4)

- However, we do not observe $\gamma_{i,k}$
  - If we observe $\gamma_{i,k} = \Pr(Z_i = k \mid \boldsymbol{X}_i)$, then we can maximize the observed data log-likelihood
  - This suggests the iterative procedure

- **E-Step**: Calculate the responsibility parameter given $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, and $\boldsymbol{\pi}$:

$$\gamma_{i,k} = \frac{\pi_k \mathcal{N}(\boldsymbol{X}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_k \mathcal{N}(\boldsymbol{X}_j; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

- **M-Step**: Maximize the log-likelihood. I.e., obtain the MLE estimator given $\gamma_{i,k}$ by

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{n} \gamma_{i,k} \boldsymbol{X}_i}{\sum_{i=1}^{n} \gamma_{i,k}}, \quad \boldsymbol{\Sigma}_k = \frac{\sum_{i=1}^{n} \gamma_{i,k} (\boldsymbol{X}_i - \boldsymbol{\mu}_k)(\boldsymbol{X}_i - \boldsymbol{\mu}_k)^\top}{\sum_{i=1}^{n} \gamma_{i,k}}$$

$$\pi_k = \frac{\sum_{i=1}^{n} \gamma_{i,k}}{N}$$

EM Algorithm in general (1)

- Let's understand the general logic and formulation for EM algorithm

- **Setup**: Let $\boldsymbol{x}$ be the observed data, $\boldsymbol{z}$ be latent, and θ be the parameter of the model
  - Then, the full-data log likelihood is written as $\log p(\boldsymbol{x}, \boldsymbol{z}, \theta)$ and observed data log likelihood (known as evidence) is $\log p(\boldsymbol{x}; \theta)$
  - Our goal is to maximize the observed log-likelihood $\log p(\boldsymbol{x}; \theta)$, but suppose that its optimization is hard while the optimization of full data likelihood is significantly easier

- Understand the difference between latent variable and parameter
  - Latent variable: random, varied by each unit
  - Parameter: fixed, shared across units

EM Algorithm in general (2)

- Now, notice that

$$\underbrace{\log p(\boldsymbol{x};\theta)}_{\text{observed}} = \log\left(\sum_{\boldsymbol{z}} \underbrace{p(\boldsymbol{x},\boldsymbol{z},\theta)}_{\text{complete}}\right)$$

  - As with the case of GMM, the observed log-likelihood is log of sum over complete likelihood, which is hard to optimize

- **Algorithm**:
  - **E-step** (Expectation step):

$$Q(\theta \mid \theta^{(t)}) = \mathbb{E}_{\boldsymbol{z}|\boldsymbol{x},\theta^{(t)}}\left[\log p(\boldsymbol{x},\boldsymbol{z};\theta)\right]$$

    - Take expectation of the **complete-data log-likelihood** with respect to the posterior of latent variables under current parameter $\theta^{(t)}$
  - **M-step** (Maximization step):

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

- **Intuition**: The difficulty comes from $\log p(\boldsymbol{x};\theta) = \log \sum_{\boldsymbol{z}} p(\boldsymbol{x},\boldsymbol{z};\theta)$
  - EM avoids this by:
    - **E-step**: "fill in" missing data using current model (soft assignment)
    - **M-step**: maximize as if data were complete

Theory of EM Algorithm (1)

- Let's understand why EM algorithm works

- Now, notice that

$$\begin{aligned}
\log p(x;\theta) &= \log\left(\sum_z p(x,z;\theta)\right) \\
&= \log\left(\sum_z \frac{p(x,z;\theta)}{q(z)} q(z)\right) \\
&\geq \sum_z \log \frac{p(x,z;\theta)}{q(z)} q(z) \quad (\because \text{Jensen's inequality}) \\
&= \mathbb{E}_z[\log p(x,z;\theta)] - \mathbb{E}_z[\log q(z)] := \mathcal{L}(q,\theta)
\end{aligned}$$

where the last term $\mathcal{L}(q,\theta)$ is called **Evidence Lower Bound** (ELBO)

Theory of EM Algorithm (2)

- Notice that ELBO is written as

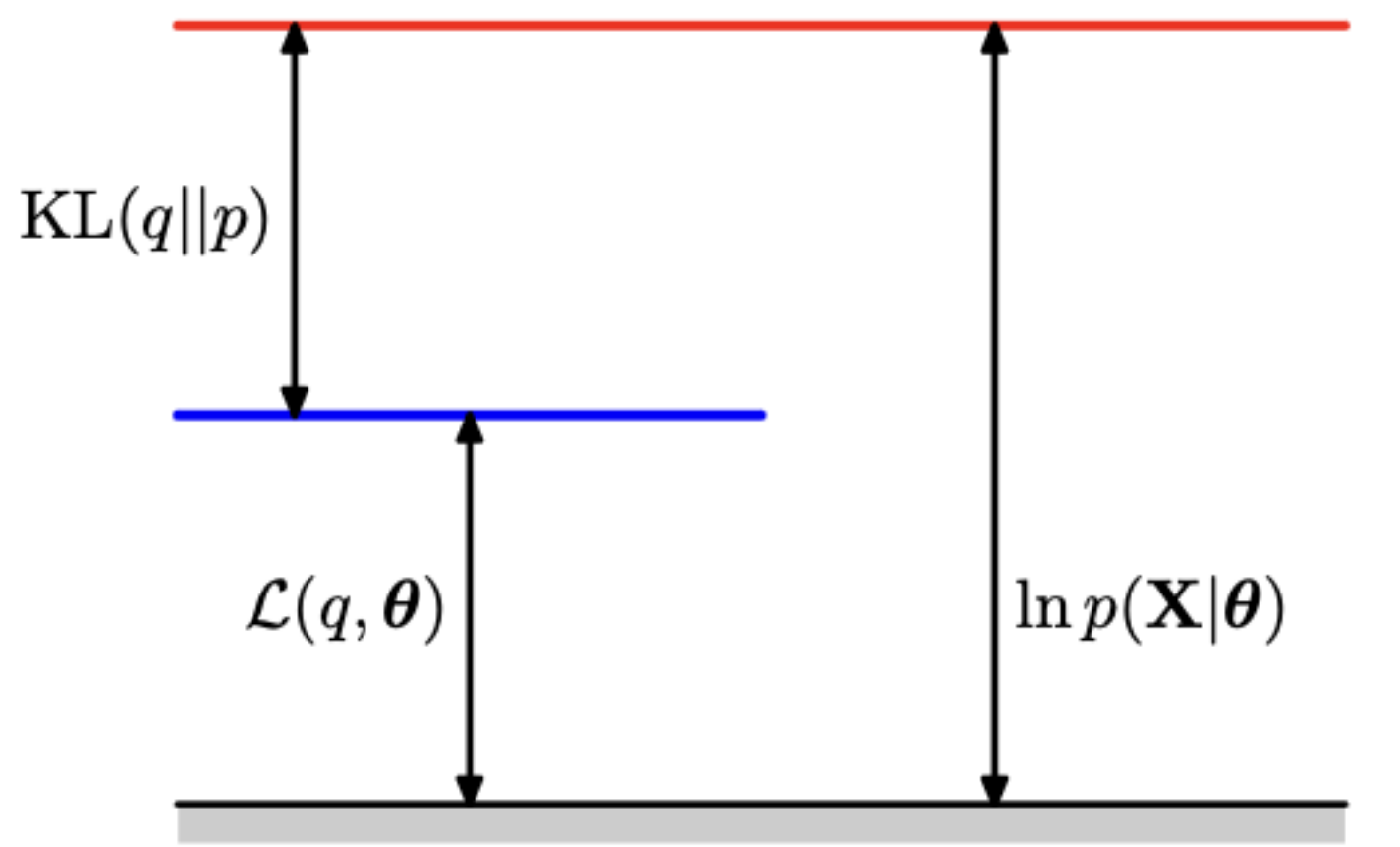
$$\begin{aligned}\mathcal{L}(q,\theta) &= \mathbb{E}_z[\log p(z \mid x;\theta)p(x;\theta)] - \mathbb{E}_z[\log q(z)] \\ &= \mathbb{E}_z[\log p(z \mid x;\theta)] - \mathbb{E}_z[\log q(z)] + \log p(x;\theta) \\ &= \underbrace{\log p(x;\theta)}_{\text{Evidence}} - \mathrm{KL}(q(z)||p(z \mid x;\theta))\end{aligned}$$

  - So, ELBO becomes lower bound when KL divergence is zero
- In other words,

$$\log p(x;\theta) = \mathrm{KL}(q(z)||p(z \mid x;\theta)) + \mathcal{L}(q,\theta)$$

- Recall that KL-divergence is non-negative and takes 0 when $q(z) = p(z \mid x;\theta)$

Theory of EM Algorithm (3)

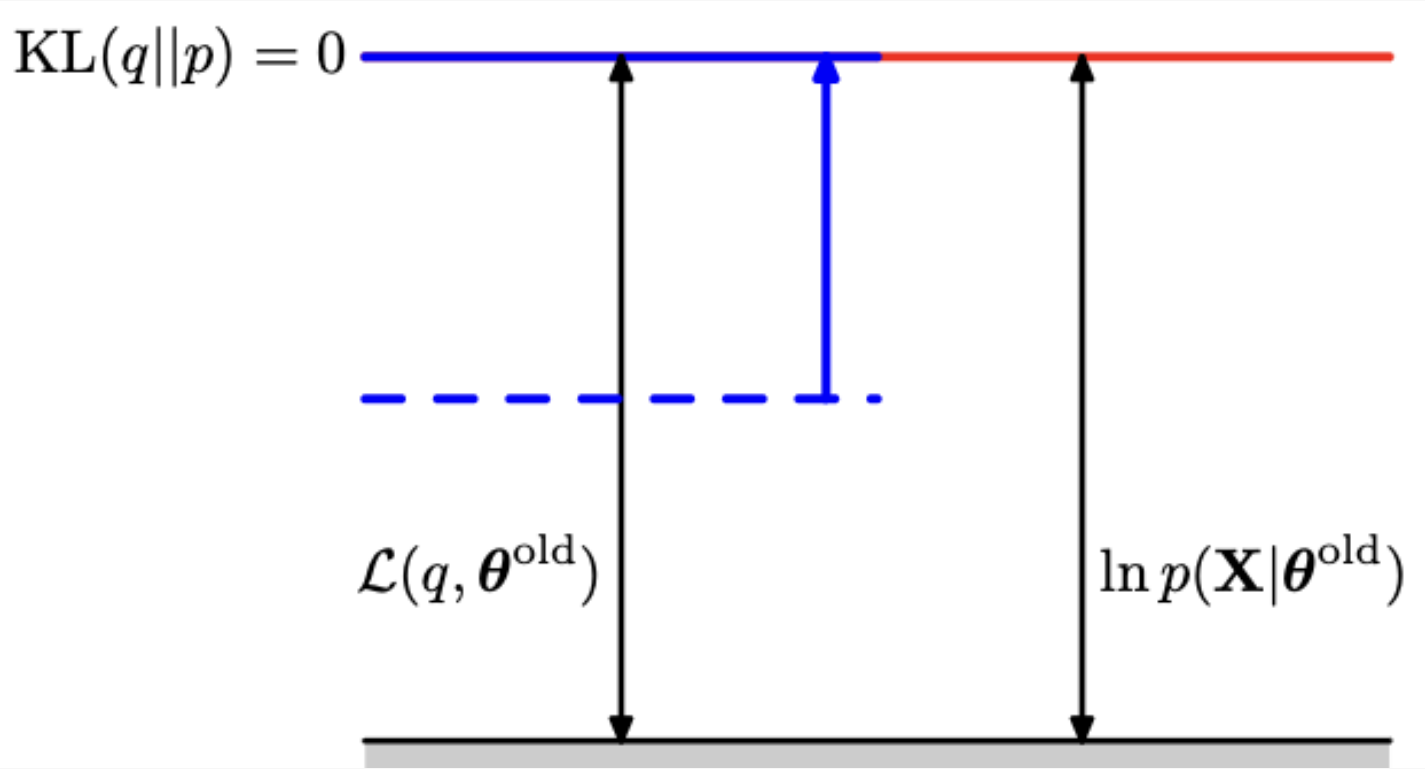


- Recall that our goal is to maximize the observed log-likelihood $\log p(x; \theta)$
  - This corresponds to maximize the lower bound $\mathcal{L}(q, \theta)$
- Notice that we have two unknowns: q and θ
  - We iteratively update them

Theory of EM Algorithm (4)

- **E-step**: Maximize ELBO $\mathcal{L}(q, \theta)$ with respect to q while fixing $\theta = \theta^{\text{old}}$.
  - Notice that once θ is fixed, we can maximize ELBO by choosing q makes KL divergence to be 0 (recall the previous derivation)
  - Such q is given by

$$q(z) = p(z \mid x; \theta)$$



Theory of EM Algorithm (5)

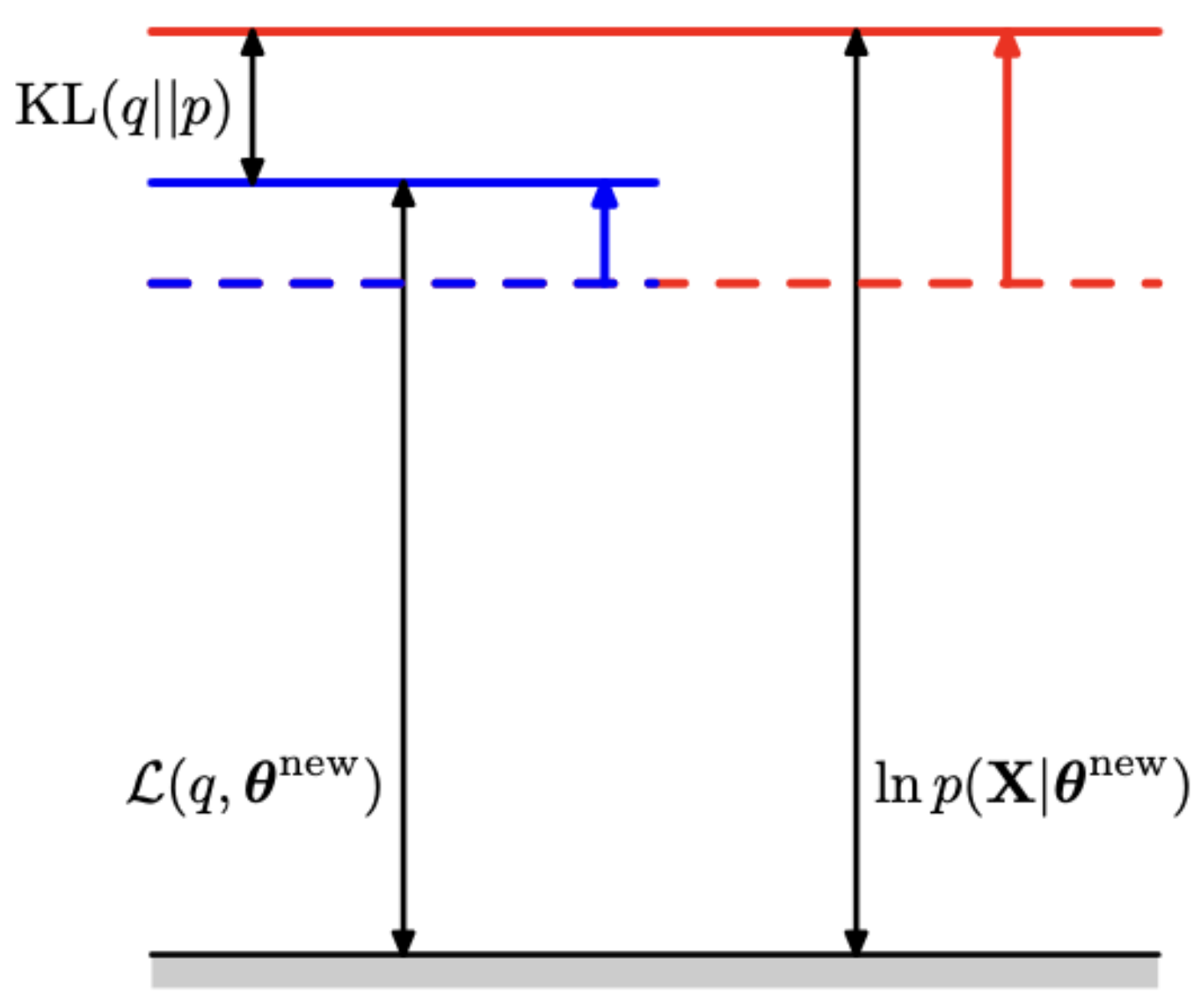
- **M-step**: Maximize ELBO $\mathcal{L}(q, \theta)$ with respect to θ while fixing q

- Notice that after E-step (setting $q(z) = p(z \mid x; \theta^{\text{old}})$), ELBO is

$$\begin{aligned}\mathcal{L}(q, \theta) &= \sum_z \log p(x, z; \theta) p(z \mid x; \theta^{\text{old}}) \\ &\qquad - \sum_z p(z \mid x; \theta^{\text{old}}) \log p(z \mid x; \theta^{\text{old}}) \\ &= \underbrace{\sum_z \log p(x, z; \theta) p(z \mid x; \theta^{\text{old}})}_{\text{Q-function}} + \text{constant}\end{aligned}$$

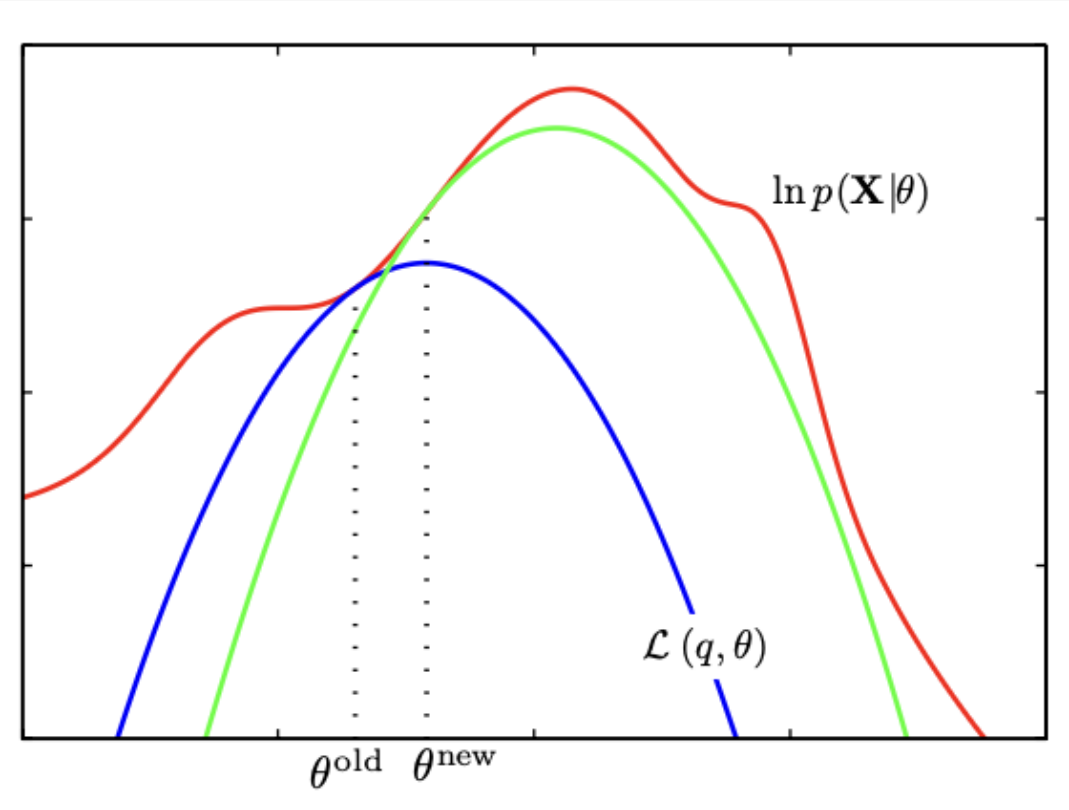
- This is the justification of maximizing Q-function

Theory of EM Algorithm (6)



Theory of EM Algorithm (7)

- EM algorithm has monotone convergence property



- E-step: Make ELBO $\mathcal{L}(q, \theta)$ tangential to log-likelihood (maximization of ELBO w.r.t. q under fixed θ)
- M-step: Make ELBO $\mathcal{L}(q, \theta)$ larger w.r.t. θ

Theory of EM Algorithm (8): Monotone convergence

- Let $q^{(t)}(\boldsymbol{z}) = p(\boldsymbol{z} \mid \boldsymbol{x}; \theta^{(t)})$

- From the ELBO decomposition,

$$\log p(\boldsymbol{x}; \theta) = \mathcal{L}(q, \theta) + \mathrm{KL}\{q(\boldsymbol{z}) \,||\, p(\boldsymbol{z} \mid \boldsymbol{x}; \theta)\}.$$

- In the E-step, we set $q^{(t)}(\boldsymbol{z}) = p(\boldsymbol{z} \mid \boldsymbol{x}; \theta^{(t)})$, so

$$\mathrm{KL}\{q^{(t)}(\boldsymbol{z}) \,||\, p(\boldsymbol{z} \mid \boldsymbol{x}; \theta^{(t)})\} = 0.$$

- Therefore,

$$\log p(\boldsymbol{x}; \theta^{(t)}) = \mathcal{L}(q^{(t)}, \theta^{(t)}).$$

Theory of EM Algorithm (9): Monotone convergence

- In the M-step, we choose

$$\theta^{(t+1)} = \arg\max_{\theta} \mathcal{L}(q^{(t)}, \theta).$$

- Therefore,

$$\mathcal{L}(q^{(t)}, \theta^{(t+1)}) \geq \mathcal{L}(q^{(t)}, \theta^{(t)}).$$

- Also, because KL divergence is non-negative,

$$\log p(\boldsymbol{x}; \theta^{(t+1)}) \geq \mathcal{L}(q^{(t)}, \theta^{(t+1)}).$$

- Combining the inequalities,

$$\log p(\boldsymbol{x}; \theta^{(t+1)}) \geq \mathcal{L}(q^{(t)}, \theta^{(t+1)}) \geq \mathcal{L}(q^{(t)}, \theta^{(t)}) = \log p(\boldsymbol{x}; \theta^{(t)}).$$

- Thus, the observed-data log-likelihood never decreases.

Uncertainty quantification of EM

- EM algorithm gives you the point estimates θ, but you cannot directly obtain the uncertainty estimates
  - Bootstrap is the easiest, but it is computationally too expensive
  - If you can work with the observed log-likelihood, then the asymptotic theory of likelihood tells us that

$$\sqrt{n}(\hat{\theta} - \theta_0) \xrightarrow{d} \mathcal{N}(0, I^{-1}), \qquad I(\theta_0) = \mathbb{E}[s_i(\theta_0) s_i(\theta_0)^\top]$$

    where $s_i(\theta_0) = \nabla \log p(x_i; \theta)$ is the score.
  - However, we often use EM because the observed log-likelihood is intractable and it is hard to compute Hessian

- Key tools: **Missing Information Principle**
  - Formally,

$$\underbrace{-\ell''(\theta \mid \boldsymbol{X})}_{\text{Observed}} = \mathbb{E}[\underbrace{-\ell''(\theta \mid \boldsymbol{X}, \boldsymbol{Z})}_{\text{Complete}} \mid \boldsymbol{X}] - \mathbb{E}[\underbrace{-\nabla^2 \log f(\boldsymbol{Z} \mid \boldsymbol{X}, \theta)}_{\text{Missing}} \mid \boldsymbol{X}]$$

  - Implication: We can easily obtain the complete data information (similar to E-step), so we only need to obtain missing data part
    - Importantly, this avoids differentiating the observed log-likelihood directly (which involves log-sum structure)

Derivation of Missing-Data Principle

- Now, by factorization,

$$f(\boldsymbol{X}, \boldsymbol{Z} \mid \theta) = f(\boldsymbol{Z} \mid \boldsymbol{X}, \theta) f(\boldsymbol{X} \mid \theta).$$

- So taking logs gives

$$\ell(\theta \mid \boldsymbol{X}, \boldsymbol{Z}) = \log f(\boldsymbol{X}, \boldsymbol{Z} \mid \theta) = \log f(\boldsymbol{Z} \mid \boldsymbol{X}, \theta) + \log f(\boldsymbol{X} \mid \theta).$$

- Now take the Hessian with respect to θ, we get

$$\ell''(\theta \mid \boldsymbol{X}) = \ell''(\theta \mid \boldsymbol{X}, \boldsymbol{Z}) - \nabla^2 \log f(\boldsymbol{Z} \mid \boldsymbol{X}, \theta).$$

- Finally, take conditional expectation given $\boldsymbol{X}$ (notice that the first term only depends on $\boldsymbol{X}$)

$$-\ell''(\theta \mid \boldsymbol{X}) = \mathbb{E}[-\ell''(\theta \mid \boldsymbol{X}, \boldsymbol{Z}) \mid \boldsymbol{X}] - \mathbb{E}\left[-\nabla^2 \log f(\boldsymbol{Z} \mid \boldsymbol{X}, \theta) \mid \boldsymbol{X}\right].$$

Extra: Approximation Methods

- **Problem of EM algorithm**: We need to evaluate the expectation of the complete-data log-likelihood with respect to the posterior distribution of latent variable
  - In many applications, however, this is not possible.

- In such situation, we need to use approximation methods
  - Stochastic: MCMC (Metropolis Hasting, Gibbs Sampling, Hamiltonian Monte Carlo)
    - Gives accurate estimates with uncertainty quantification, but not scalable for large and complex data
  - Deterministic: Variational Inference
    - Gives approximate estimates, but scalable and fast

- This is not required for final exam, but it is the main technique you need to use in future
  - Variational Inference: Ch.10 of Bishop (2006), Blei et al. (2017 JASA)
  - MCMC: Ch.11 of Bishop (2006)

Extra: Variational Inference (1)

- Now, we are in the bayesian framework. So, the parameter θ is a random variable
  - Thus, we denote the set of all parameters and latent variable as $\boldsymbol{Z}$
  - Our goal is to find the distribution that is the closest to the exact posterior $p(\boldsymbol{Z} \mid \boldsymbol{X})$, which is minimizing the KL-divergence $\mathrm{KL}(q(z) \mid p(z \mid x))$ over some candidate set of distributions $\mathcal{Q}$.

- Importantly, the minimization of KL divergence is equivalent to the maximization of ELBO:

$$
\begin{aligned}
q^*(z) &= \arg\min_{q(z)\in\mathcal{Q}} \mathrm{KL}(q(z) \,||\, \underbrace{p(z \mid x)}_{\text{Posterior}}) \\
&= \arg\min_{q(z)\in\mathcal{Q}} \left(\underbrace{\log p(x)}_{\text{constant}} - \mathrm{ELBO}(q) \right) \\
&= \arg\max_{q(z)\in\mathcal{Q}} \mathrm{ELBO}(q)
\end{aligned}
$$

Extra: Variational Inference (2)

- What kind of $\mathcal{Q}$ should we use?
  - **Mean Field Approximation**: Choose $\mathcal{Q}$ to restrict that latent variables are mutually independent; i.e.,

$$q(z) = \prod_{j=1}^{m} q_j(z_j)$$

  - It is crazy assumption, but often works well in practice

- **Pro**: Mean-field family is expressive because it can capture any marginal density of the latent variable
- **Con**: It cannot capture the correlation between latent variables

Extra: Variational Inference (3)

- Now, let's focus on j-th latent $q_j(z_j)$.

$$
\begin{aligned}
&\mathrm{ELBO}(q_j) \\
&= \mathbb{E}_{z_j \sim q_j}\Bigg[\underbrace{\mathbb{E}_{z_{-j}}\Big\{\log p(x, z_j, z_{-j})\Big\}}_{:=f(z_j)}\Bigg] - \mathbb{E}_{z_j \sim q_j}[\log q_j(z_j)] + \mathrm{const} \\
&= -\mathbb{E}_{z_j \sim q_j}\left[\log \frac{q_j(z_j)}{\exp(f(z_j))}\right] + \mathrm{const} \\
&= -\mathrm{KL}(q_j(z_j) || \exp(f(z_j))) + \mathrm{const}
\end{aligned}
$$

- Thus, when ELBO is maximized,

$$
q_j^*(z_j) \propto \exp\Big\{\mathbb{E}_{z_{-j}}\Big\{\log p(x, z_j, z_{-j})\Big\}\Big\}
$$

Extra: Variational Inference (4)

Coordinate ascent mean-field variational inference (CAVI)

- **Input**: A model $p(x, z)$, data $p(z)$
- **Output**: Variational density $q(z)$
- **Algorithm**

  1. for each $j \in 1, \cdots, m$, set

$$q_j(z_j) \propto \exp\left\{ \mathbb{E}_{z_{-j}} \left\{ \log p(x, z_j, z_{-j}) \right\} \right\}$$

     - Each coordinate density is conditional on all the other coordinates
     - Similar to Gibbs sampling

  2. Compute $\mathrm{ELBO}(q)$ and repeat until convergence

- **Caveat**: ELBO is a non-convex objective function, which is only guaranteed to converge to a local optimum

$\rightarrow$ Sensitive to the initialization

Extra: From VI to MCMC

- In both VI and MCMC, the goal is often to approximate a complicated distribution
- In Bayesian latent variable models, this target is usually the posterior

$$p^*(\boldsymbol{z}) = p(\boldsymbol{z} \mid \boldsymbol{X}).$$

- But for MCMC theory, we do not need to write the observed data $\boldsymbol{X}$ explicitly
- So, from now on, we simply write the target distribution as

$$p^*(\boldsymbol{z}).$$

- **VI**: approximate $p^*(\boldsymbol{z})$ by optimizing over $q(\boldsymbol{z})$
- **MCMC**: generate samples from $p^*(\boldsymbol{z})$ by constructing a Markov chain whose stationary distribution is $p^*(\boldsymbol{z})$

Extra: Theory of MCMC (1)

- A first-order Markov chain is defined as

$$p(\boldsymbol{z}^{m+1} \mid \boldsymbol{z}^1, \cdots, \boldsymbol{z}^m) = p(\boldsymbol{z}^{m+1} \mid \boldsymbol{z}^m)$$

- Thus, by the law of total probability,

$$p(\boldsymbol{z}^{m+1}) = \sum_{\boldsymbol{z}^m} \underbrace{p(\boldsymbol{z}^{m+1} \mid \boldsymbol{z}^m)}_{\text{Transition Prob}} p(\boldsymbol{z}^m)$$

- **Detailed Balancing**: With transition probabilities $T(\boldsymbol{z}, \boldsymbol{z}')$,

$$p^*(\boldsymbol{z}) T(\boldsymbol{z}, \boldsymbol{z}') = p^*(\boldsymbol{z}') T(\boldsymbol{z}', \boldsymbol{z})$$

- Detailed balancing is a *sufficient* condition to ensure that the required distribution $p(\boldsymbol{z})$ is stationary

Extra: Theory of MCMC (2)

- The detailed balancing implies stationarity because of the following:

$$
\begin{aligned}
p(z) &= \sum_{z'} p^{prev}(z')\, T(z', z) \\
&= \sum_{z'} p^{prev}(z)\, T(z, z') \quad \text{(detailed balancing)} \\
&= p^{prev}(z) \sum_{z'} T(z, z') \\
&= p^{prev}(z) \underbrace{\sum_{z'} p(z' \mid z)}_{=1} \\
&= p^{prev}(z)
\end{aligned}
$$

Extra: Theory of MCMC (3)

- We want our chains to converge (in the limit) to the same stationary state, regardless of starting distribution.
    - Such chains are called **ergodic**.

- For finite state spaces, our chains converge to equilibrium under two relatively weak conditions
    1. **Irreducibility**: Any set of states can be reached from any other state in a finite number of moves $\rightarrow$ Stationary distribution is unique
    2. **Aperiodicity**: The chain does not get trapped in cycles

- Rarely justified theoretically
    - Instead, we can check it empirically by running the estimation multiple times (chains) and checking the convergence (e.g., trace plot / Rhat)

Extra: Metropolis-Hasting (M-H) algorithm

1. **Initialization**: Choose an arbitrary point x'
2. For each iteration at step m,
   - 2.1 **Propose** a candidate z^* for the next sample by picking from a proposal distribution $q(z^* \mid z^m)$
   - 2.2 **Accept** the sample if $u \leq \alpha(z^*, z^m)$, where $u \sim \mathrm{Unif}(0, 1)$ and

$$\alpha(z^*, z^m) = \min\left(1, \frac{\tilde{p}(z^*)q(z^m \mid z^*)}{\tilde{p}(z^m)q(z^* \mid z^m)}\right)$$

- This works because it satisfies the detailed balancing:

$$p(\boldsymbol{z})q(\boldsymbol{z} \mid \boldsymbol{z}')\alpha(\boldsymbol{z}', \boldsymbol{z}) = \min\Big\{p(\boldsymbol{z})q(\boldsymbol{z} \mid \boldsymbol{z}'), p(\boldsymbol{z}')q(\boldsymbol{z}' \mid \boldsymbol{z})\Big\}$$

$$= \min\Big\{p(\boldsymbol{z}')q(\boldsymbol{z}' \mid \boldsymbol{z}), p(\boldsymbol{z})q(\boldsymbol{z} \mid \boldsymbol{z}')\Big\} = p(\boldsymbol{z}')q(\boldsymbol{z}' \mid \boldsymbol{z})\alpha(\boldsymbol{z}, \boldsymbol{z}')$$

Extra: Metropolis algorithm

- Suppose that your proposal distribution is symmetric: i.e.,

$$q(y \mid x) = q(x \mid y)$$

  - Example: $q(x \mid y) = q((x - y)^2)$
- Then, your ratio is

$$\alpha(z^*, z^m) = \min\left(1, \frac{\tilde{p}(z^*)}{\tilde{p}(z^m)}\right)$$

- This is called **Metropolis algorithm**
- **Problem (both M and M-H)**: The acceptance rate may be low, sensitive to the choice of $q(\cdot \mid \cdot)$, does not work well for high-dimensional settings.

Extra: Gibbs Sampling

- **Setting**: Suppose we want to get $p(z_1, \cdots, z_k)$
- **Algorithm**: After initialization, for each step m, keep sampling

$$
\begin{aligned}
z_1^{m+1} &\sim p(z_1 \mid z_2^m, \cdots, z_k^m) \\
z_2^{m+1} &\sim p(z_2 \mid z_1^{m+1}, z_3^m, \cdots, z_k^m) \\
&\vdots \\
z_k^{m+1} &\sim p(z_k \mid z_1^{m+1}, z_2^{m+1}, \cdots, z_{k-1}^{m+1})
\end{aligned}
$$

- **AD**: Always accept the sample
- **DA**: This approach works only when full conditional distributions are easy to sample

Extra: Gibbs Sampling as Special Case of M-H

- If we use the proposal $q(z^* \mid z^m) = p(z_i^* \mid z_{-i}^m)$ and $q(z^m \mid z^*) = p(z_i^m \mid z_{-i}^*)$, then Gibbs Sampling can be seen as a special case of M-H
- In Gibbs Sampling, we always accept the proposal because

$$\begin{aligned}
\alpha(z^*, z^m) &= \min\left(1, \frac{\tilde{p}(z^*)q(z^m \mid z^*)}{\tilde{p}(z^m)q(z^* \mid z^m)}\right) \quad \text{(Definition)} \\
&= \min\left(1, \frac{\tilde{p}(z_{-i}^*)\tilde{p}(z_i^* \mid z_{-i}^*)p(z_i^m \mid z_{-i}^*)}{p(z_{-i}^m)\tilde{p}(z_i^m \mid z_{-i}^m)p(z_i^* \mid z_{-i}^m)}\right) \\
&= \min(1, 1) = 1
\end{aligned}$$

where the last equality is by $z_{-i}^* = z_{-i}^m$ (as only i changes)

Extra: Metropolis within Gibbs

- Gibbs sampling is sometimes hard to implement when full-conditional is not known
  - M-H can solve this problem, but does not work well for high-dimensional setting
  - This is because M-H considers the update of all the parameters at the same time as marginal

- We instead consider doing the M-H for each full conditional in Gibbs sampling

- This is called **Metropolis within Gibbs**

Extra: Collapsed Gibbs Sampling

- We can speed up the Gibbs sampler by marginalizing some of the latent variables.
  - This is called **Collapsed Gibbs Sampler**
  - **AD**: Faster convergence (b/c fewer parameters to be sampled)
- **Setup**: let's split the latent variable into
  1. z: variables to sample
  2. θ: variables to collapse
- Then, instead of sample from $p(z, \theta)$, we can integrate out θ:

$$p(z \mid x) = \int p(z, \theta \mid x) d\theta$$

Then, we run Gibbs sampler updates like:

$$z_i^{(t+1)} \sim p(z_i \mid z_{-i}, x)$$

under the marginal posterior

Extra: Hamiltonian/Hybrid Monte Carlo (HMC)

- **Idea**: To use momentum to update the states to mitigate the random walk behavior of GS and MH:
  - x: position variable / u: momentum variable
  - Consider

$$\underbrace{H(x_i, u_i)}_{\text{Hamiltonian}} = \underbrace{U(x_i)}_{\text{Potential Energy}} + \underbrace{K(u_i)}_{\text{Kinetic Energy}} = -\log p(x) + \frac{1}{2} u^T M^{-1} u$$

  which satisfies Hamiltonian Equations:

$$\frac{\partial x}{\partial t} = \frac{\partial H}{\partial u}, \quad \frac{\partial u}{\partial t} = -\frac{\partial H}{\partial x}$$

Extra: Hamiltonian/Hybrid Monte Carlo (HMC)

- For each iteration at step m,
  1. Updating momentum distribution $u_0 \sim N(0, M)$
  2. For $l = 1, \cdots, L-1$, updating (x, u) by

$$u^{l+1/2} \leftarrow u^l + \frac{1}{2}\epsilon \frac{d \log(x \mid y)}{dx}$$
$$x^{l+1} \leftarrow x^l + \epsilon M^{-1} u^l$$
$$u^{l+1} \leftarrow u^{l+1/2} + \frac{1}{2}\epsilon \frac{d \log(x \mid y)}{dx}$$

  3. Accept (u^L, x^L) with probability

$$\min\left(\frac{p(x^* \mid y)p(u^*)}{p(x^{t-1} \mid y)p(u^{t-1})}, 1\right)$$

- **Takeaway**: HMC is one type of M-H methods!
  - Notice that we accept or reject in the same way as M-H