Lecture 22: Spectral clustering, EM algorithm

Nisha Chandramoorthy

November 14, 2023

Lloyd's algorithm

- Randomly choose k centers $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$.

Lloyd's algorithm

- Randomly choose k centers $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$.
- Given centers $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$, assign each point x_i to the closest center. That is,

$$C_j = \{x_i : j \in \operatorname{argmin}_l \|x_i - \mu_l\|\}.$$

Lloyd's algorithm

- Randomly choose k centers $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$.
- Given centers $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$, assign each point x_i to the closest center. That is,

$$C_j = \{x_i : j \in \operatorname{argmin}_l \|x_i - \mu_l\|\}.$$

- Given clusters $C_1, \ldots, C_k$, update centers $\mu_1, \ldots, \mu_k \in \mathbb{R}^d$ as

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i.$$

k-means algorithm (Lloyd's algorithm)

- Lloyd's algorithm is an approximate method to solve the ERM problem:

$$\min_{C_1,\ldots,C_k} \sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - \mu(C_j)\|^2.$$

k-means algorithm (Lloyd's algorithm)

- Lloyd's algorithm is an approximate method to solve the ERM problem:

$$\min_{C_1,\ldots,C_k} \sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - \mu(C_j)\|^2.$$

- here, $\mu(C_j) = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i = \operatorname{argmin}_{\mu \in \mathbb{R}^d} \sum_{x_i \in C_j} \|x_i - \mu\|^2$ is the mean of the points in cluster C_j.

k-means algorithm (Lloyd's algorithm)

- Lloyd's algorithm is an approximate method to solve the ERM problem:

$$\min_{C_1,\dots,C_k} \sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - \mu(C_j)\|^2.$$

- here, $\mu(C_j) = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i = \operatorname{argmin}_{\mu \in \mathbb{R}^d} \sum_{x_i \in C_j} \|x_i - \mu\|^2$ is the mean of the points in cluster C_j.
- Lloyd's algorithm is a heuristic. It is not guaranteed to converge to the global optimum or even a local minimum.

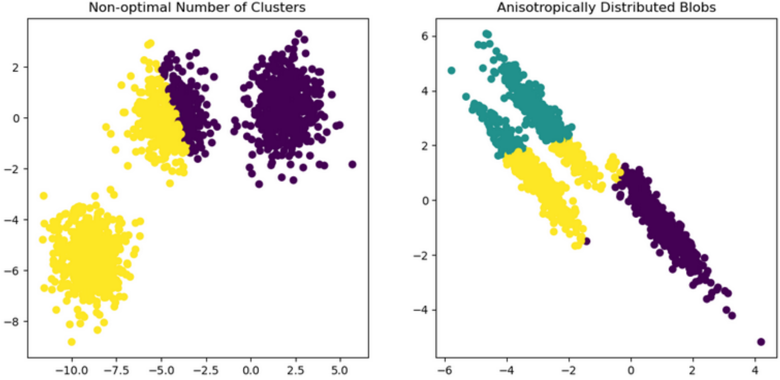
Lloyd's algorithm properties

- k-means algorithm is sensitive to initialization of the centers.

Lloyd's algorithm properties

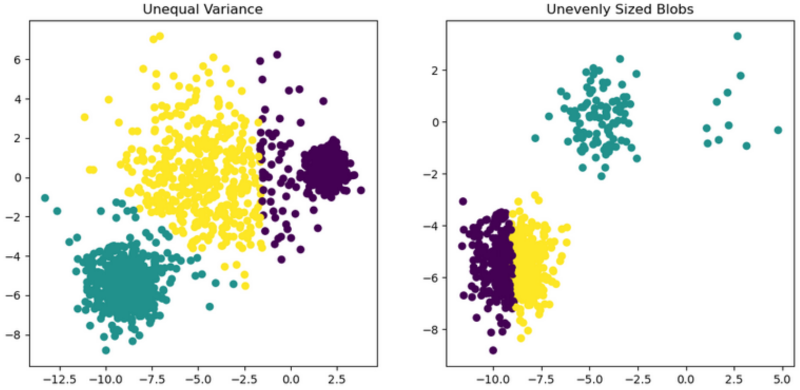
- k-means algorithm is sensitive to initialization of the centers.
- Complexity: $O(mdk)$ per iteration, where m is the number of points, d is the dimension, and k is the number of clusters.

k-means failure modes



Source: sklearn's toy examples

k-means failure modes contd



Source: sklearn's toy examples

Spectral clustering

- Given distance d or similarity matrix, $W \in \mathbb{R}^{m \times m}$, partition the points into k clusters.

Spectral clustering

- Given distance d or similarity matrix, $W \in \mathbb{R}^{m \times m}$, partition the points into k clusters.
- W is symmetric and non-negative.

Spectral clustering

- Given distance d or similarity matrix, $W \in \mathbb{R}^{m \times m}$, partition the points into k clusters.
- W is symmetric and non-negative.
- W is a weighted adjacency matrix of a graph.

Spectral clustering

- Given distance d or similarity matrix, $W \in \mathbb{R}^{m \times m}$, partition the points into k clusters.
- W is symmetric and non-negative.
- W is a weighted adjacency matrix of a graph.
- ERM problem: $\min_{C_1,\ldots,C_k} \sum_{j=1}^{k} \sum_{x_i \in C_j} \sum_{x_l \notin C_j} w_{il}$. Graph min-cut problem.

RatioCut problem: spectral clustering solution

- RatioCut problem: $\min_{C_1,\dots,C_k} \sum_{j=1}^{k} \frac{\sum_{x_i \in C_j} \sum_{x_l \notin C_j} w_{il}}{|C_j|}$.

RatioCut problem: spectral clustering solution

- RatioCut problem: $\min_{C_1,\ldots,C_k} \sum_{j=1}^{k} \frac{\sum_{x_i \in C_j} \sum_{x_l \notin C_j} w_{il}}{|C_j|}$.
- Normalization by $|C_j|$ penalizes small clusters.

RatioCut problem: spectral clustering solution

- RatioCut problem: $\min_{C_1,\ldots,C_k} \sum_{j=1}^{k} \frac{\sum_{x_i \in C_j} \sum_{x_l \notin C_j} w_{il}}{|C_j|}$.
- Normalization by $|C_j|$ penalizes small clusters.

RatioCut objective

- Lemma 22.3 (Ben-David and Shalev Shwartz) RatioCut objective = $\mathrm{Tr}(H^\top LH)$

RatioCut objective

- Lemma 22.3 (Ben-David and Shalev Shwartz) RatioCut objective = $\mathrm{Tr}(H^\top LH)$
- $L = D - W$ is the graph Laplacian, where D is the diagonal matrix with $D_{ii} = \sum_{j=1}^{m} w_{ij}$.

RatioCut objective

- Lemma 22.3 (Ben-David and Shalev Shwartz) RatioCut objective = $\mathrm{Tr}(H^\top LH)$
- $L = D - W$ is the graph Laplacian, where D is the diagonal matrix with $D_{ii} = \sum_{j=1}^{m} w_{ij}$.
- $H \in \mathbb{R}^{m\times k}$ is the indicator matrix of the clusters. $H_{ij} = 1i/\sqrt{|C_j|}$ if $x_i \in C_j$ and 0 otherwise.

RatioCut objective

- Lemma 22.3 (Ben-David and Shalev Shwartz) RatioCut objective = $\mathrm{Tr}(H^\top LH)$
- $L = D - W$ is the graph Laplacian, where D is the diagonal matrix with $D_{ii} = \sum_{j=1}^{m} w_{ij}$.
- $H \in \mathbb{R}^{m\times k}$ is the indicator matrix of the clusters. $H_{ij} = 1i/\sqrt{|C_j|}$ if $x_i \in C_j$ and 0 otherwise.
- h_i (ith column of H) is nonzero at row j if x_j is in cluster i.

RatioCut objective

- Lemma 22.3 (Ben-David and Shalev Shwartz) RatioCut objective = $\mathrm{Tr}(H^\top LH)$
- $L = D - W$ is the graph Laplacian, where D is the diagonal matrix with $D_{ii} = \sum_{j=1}^{m} w_{ij}$.
- $H \in \mathbb{R}^{m\times k}$ is the indicator matrix of the clusters. $H_{ij} = 1i/\sqrt{|C_j|}$ if $x_i \in C_j$ and 0 otherwise.
- h_i (ith column of H) is nonzero at row j if x_j is in cluster i.
- H has orthonormal columns.

Recall: graphical representation of X

- Choose weighting, such as, $w_{ij} = \exp(-\|x_i - x_j\|^2/2\sigma^2)$. As $\sigma \to 0$, $w_{ij} \to \mathbb{1}_{i=j}$. The $m \times m$ matrix W is the adjacency matrix of a graph.

Recall: graphical representation of X

- Choose weighting, such as, $w_{ij} = \exp(-\|x_i - x_j\|^2/2\sigma^2)$. As $\sigma \to 0$, $w_{ij} \to \mathbb{1}_{i=j}$. The $m \times m$ matrix W is the adjacency matrix of a graph.
- Let D be the diagonal matrix with $D_{ii} = \sum_{j=1}^{m} w_{ij}$.

Recall: graphical representation of X

- Choose weighting, such as, $w_{ij} = \exp(-\|x_i - x_j\|^2/2\sigma^2)$. As $\sigma \to 0$, $w_{ij} \to \mathbb{1}_{i=j}$. The $m \times m$ matrix W is the adjacency matrix of a graph.
- Let D be the diagonal matrix with $D_{ii} = \sum_{j=1}^{m} w_{ij}$.
- Graph laplacian: $L = D - W$.

Recall: graphical representation of X

- Choose weighting, such as, $w_{ij} = \exp(-\|x_i - x_j\|^2/2\sigma^2)$. As $\sigma \to 0$, $w_{ij} \to \mathbb{1}_{i=j}$. The $m \times m$ matrix W is the adjacency matrix of a graph.
- Let D be the diagonal matrix with $D_{ii} = \sum_{j=1}^{m} w_{ij}$.
- Graph laplacian: $L = D - W$.
- Detects local structure / clusters in data.

Lemma proof: RatioCut objective and graph laplacian connection

- RatioCut objective$(C_1, \cdots, C_k)$

$$:= \sum_{j=1}^{k} \frac{\sum_{x_i \in C_j} \sum_{x_l \notin C_j} w_{il}}{|C_j|}.$$

Lemma proof: RatioCut objective and graph laplacian connection

- RatioCut objective$(C_1, \cdots, C_k)$

$$:= \sum_{j=1}^{k} \frac{\sum_{x_i \in C_j} \sum_{x_l \notin C_j} w_{il}}{|C_j|}.$$

- Need to show equal to $\mathrm{Tr}(H^\top LH)$.

Laplacian eigenmaps

- Want to solve: $\min_{y_1,\cdots,y_m} \sum_{i=1}^{m} \sum_{j=1}^{m} w_{ij} \|y_i - y_j\|^2$.

Laplacian eigenmaps

- Want to solve: $\min_{y_1,\cdots,y_m} \sum_{i=1}^{m} \sum_{j=1}^{m} w_{ij} \|y_i - y_j\|^2$.
- optimal embeddings: $y_i = E(x_i) = U[i, -n:]$ where U is the matrix of eigenvectors of L.

Laplacian eigenmaps

- Want to solve: $\min_{y_1,\cdots,y_m} \sum_{i=1}^m \sum_{j=1}^m w_{ij} \|y_i - y_j\|^2$.
- optimal embeddings: $y_i = E(x_i) = U[i, -n:]$ where U is the matrix of eigenvectors of L.
- For any vector v, $v^\top L v = (1/2) \sum_{i,j=1}^m w_{ij}(v_i - v_j)^2$.

Laplacian eigenmaps

- Want to solve: $\min_{y_1,\cdots,y_m} \sum_{i=1}^{m} \sum_{j=1}^{m} w_{ij} \|y_i - y_j\|^2$.
- optimal embeddings: $y_i = E(x_i) = U[i, -n:]$ where U is the matrix of eigenvectors of L.
- For any vector v, $v^\top L v = (1/2) \sum_{i,j=1}^{m} w_{ij}(v_i - v_j)^2$.
- L is positive semi-definite.

Bottom n eigenvectors

- Rayleigh quotient optimality

Bottom n eigenvectors

- Rayleigh quotient optimality
- Another interpretation: top n eigenvectors of $L^\dagger$. $L^\dagger_{ij}$ represents expected time for random walk $i \to j \to i$.

Bottom n eigenvectors

- Rayleigh quotient optimality
- Another interpretation: top n eigenvectors of $L^\dagger$. $L^\dagger_{ij}$ represents expected time for random walk $i \to j \to i$.
- Kernel PCA with $K = L^\dagger$ is equivalent to Laplacian eigenmaps.

Combining dimension reduction and k-means

- Spectral clustering algorithm uses Laplacian eigenmaps on m-dimensional data.

Combining dimension reduction and k-means

- Spectral clustering algorithm uses Laplacian eigenmaps on m-dimensional data.
- Uses $v_i, i = 1, 2, \cdots, k$ eigenvectors of L corresponding to the k smallest eigenvalues.

Combining dimension reduction and k-means

- Spectral clustering algorithm uses Laplacian eigenmaps on m-dimensional data.
- Uses $v_i, i = 1, 2, \cdots, k$ eigenvectors of L corresponding to the k smallest eigenvalues.
- Perform k-means on rows of v_i. to obtain clusters

Gaussian mixtures

- Suppose we want to cluster data that is generated from a mixture of Gaussians.

Gaussian mixtures

- Suppose we want to cluster data that is generated from a mixture of Gaussians.
- $x_i \sim \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$.

Gaussian mixtures

- Suppose we want to cluster data that is generated from a mixture of Gaussians.
- $x_i \sim \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$.
- Frequentist view: there is a true (unknown) parameter $\theta = (\pi_1, \ldots, \pi_k, \mu_1, \ldots, \mu_k, \Sigma_1, \ldots, \Sigma_k)$ that generated the data.

- Clustering objective: maximize log likelihood of the data.

- Clustering objective: maximize log likelihood of the data.
- $\ell(x, \theta) = \log p_\theta(x) = \log \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$.

- Clustering objective: maximize log likelihood of the data.
- $\ell(x, \theta) = \log p_\theta(x) = \log \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$.
- $\hat{R}_S(\theta) = \sum_{i=1}^{m} \log \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$.

- Clustering objective: maximize log likelihood of the data.
- $\ell(x, \theta) = \log p_\theta(x) = \log \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$.
- $\hat{R}_S(\theta) = \sum_{i=1}^{m} \log \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$.
- More generally, $\hat{R}_S(\theta) = \sum_{i=1}^{m} \log \sum_{j=1}^{k} q_\theta(z_j) p_\theta(x_i|z_j)$.

- Clustering objective: maximize log likelihood of the data.
- $\ell(x, \theta) = \log p_\theta(x) = \log \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$.
- $\hat{R}_S(\theta) = \sum_{i=1}^{m} \log \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$.
- More generally, $\hat{R}_S(\theta) = \sum_{i=1}^{m} \log \sum_{j=1}^{k} q_\theta(z_j) p_\theta(x_i | z_j)$.
- The joint distribution $p_\theta(x, z) = q_\theta(z) p_\theta(x|z)$ is parametrized by θ.

- Clustering objective: maximize log likelihood of the data.
- $\ell(x, \theta) = \log p_\theta(x) = \log \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$.
- $\hat{R}_S(\theta) = \sum_{i=1}^{m} \log \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$.
- More generally, $\hat{R}_S(\theta) = \sum_{i=1}^{m} \log \sum_{j=1}^{k} q_\theta(z_j) p_\theta(x_i | z_j)$.
- The joint distribution $p_\theta(x, z) = q_\theta(z) p_\theta(x|z)$ is parametrized by θ.
- Z is a latent variable, e.g., Z is the cluster assignment of X.

Maximizing log likelihood

- Distribution q of the latent variable is unknown.

Maximizing log likelihood

- Distribution q of the latent variable is unknown.
- Thus, we want to solve:

$$\max_{\theta} \max_{q} \sum_{i=1}^{m} \log \sum_{j=1}^{k} q_\theta(z_j) p_\theta(x_i|z_j). \tag{1}$$

- Lemma: For fixed θ, optimal $q_\theta \equiv p_\theta(X|Z)$ is the posterior distribution of Z given X.

Proof: derivation of ELBO

- Fix some x and θ.

Proof: derivation of ELBO

- Fix some x and θ.
- $\ell(x, \theta) = \log p_\theta(x) = \log \sum_{j=1}^{k} p_\theta(x, z_j) = \log \sum_{j=1}^{k} q_\theta(z_j) \frac{p_\theta(x, z_j)}{q_\theta(z_j)}$.

Proof: derivation of ELBO

- Fix some x and θ.
- $\ell(x, \theta) = \log p_\theta(x) = \log \sum_{j=1}^{k} p_\theta(x, z_j) = \log \sum_{j=1}^{k} q_\theta(z_j) \frac{p_\theta(x, z_j)}{q_\theta(z_j)}$.
- Use Jensen's inequality: $E \log Z \leqslant \log EZ$ for any random variable Z.

Proof: derivation of ELBO

- Fix some x and θ.
- $\ell(x, \theta) = \log p_\theta(x) = \log \sum_{j=1}^{k} p_\theta(x, z_j) = \log \sum_{j=1}^{k} q_\theta(z_j) \frac{p_\theta(x, z_j)}{q_\theta(z_j)}$.
- Use Jensen's inequality: $E \log Z \leqslant \log EZ$ for any random variable Z.
- Thus, $\ell(x, \theta) \geqslant \sum_{j=1}^{k} q_\theta(z_j) \log \frac{p_\theta(x, z_j)}{q_\theta(z_j)}$.

Proof: derivation of ELBO

- Fix some x and θ.
- $\ell(x,\theta) = \log p_\theta(x) = \log \sum_{j=1}^{k} p_\theta(x, z_j) = \log \sum_{j=1}^{k} q_\theta(z_j) \frac{p_\theta(x,z_j)}{q_\theta(z_j)}$.
- Use Jensen's inequality: $E \log Z \leqslant \log EZ$ for any random variable Z.
- Thus, $\ell(x,\theta) \geqslant \sum_{j=1}^{k} q_\theta(z_j) \log \frac{p_\theta(x,z_j)}{q_\theta(z_j)}$.
- This holds for any probability distribution q_θ.

Proof: derivation of ELBO

- Fix some x and θ.
- $\ell(x, \theta) = \log p_\theta(x) = \log \sum_{j=1}^{k} p_\theta(x, z_j) = \log \sum_{j=1}^{k} q_\theta(z_j) \frac{p_\theta(x, z_j)}{q_\theta(z_j)}$.
- Use Jensen's inequality: $E \log Z \leqslant \log EZ$ for any random variable Z.
- Thus, $\ell(x, \theta) \geqslant \sum_{j=1}^{k} q_\theta(z_j) \log \frac{p_\theta(x, z_j)}{q_\theta(z_j)}$.
- This holds for any probability distribution q_θ.
- $\text{ELBO}(q, \theta) = \sum_{j=1}^{k} q(z_j) \log \frac{p_\theta(x, z_j)}{q(z_j)}$.

Proof: derivation of ELBO

- Fix some x and θ.
- $\ell(x,\theta) = \log p_\theta(x) = \log \sum_{j=1}^{k} p_\theta(x, z_j) = \log \sum_{j=1}^{k} q_\theta(z_j) \frac{p_\theta(x,z_j)}{q_\theta(z_j)}$.
- Use Jensen's inequality: $E \log Z \leqslant \log EZ$ for any random variable Z.
- Thus, $\ell(x,\theta) \geqslant \sum_{j=1}^{k} q_\theta(z_j) \log \frac{p_\theta(x,z_j)}{q_\theta(z_j)}$.
- This holds for any probability distribution q_θ.
- $\mathrm{ELBO}(q,\theta) = \sum_{j=1}^{k} q(z_j) \log \frac{p_\theta(x,z_j)}{q(z_j)}$.
- Thus, we have shown, $\ell(x,\theta) \geqslant \mathrm{ELBO}(q,\theta)$ for any q.

- $\text{ELBO}(q, \theta) = -D_{KL}(q(z) || p_\theta(z|x)) + \log p_\theta(x)$.

- $\mathrm{ELBO}(q, \theta) = -D_{KL}(q(z) || p_\theta(z|x)) + \log p_\theta(x)$.
- When $q(z) = p_\theta(z|x)$, $\mathrm{ELBO}(q, \theta) = \log p_\theta(x)$.

- $\text{ELBO}(q, \theta) = -D_{KL}(q(z)||p_\theta(z|x)) + \log p_\theta(x)$.
- When $q(z) = p_\theta(z|x)$, $\text{ELBO}(q, \theta) = \log p_\theta(x)$.
- $\text{ELBO}(p_\theta(Z|X), \theta) = \sum_{j=1}^{k} p_\theta(z_j|x) \log \frac{p_\theta(z_j|x) p_\theta(x)}{p_\theta(z_j|x)}$.

- $\mathrm{ELBO}(q, \theta) = -D_{KL}(q(z) || p_\theta(z|x)) + \log p_\theta(x)$.
- When $q(z) = p_\theta(z|x)$, $\mathrm{ELBO}(q, \theta) = \log p_\theta(x)$.
- $\mathrm{ELBO}(p_\theta(Z|X), \theta) = \sum_{j=1}^{k} p_\theta(z_j|x) \log \frac{p_\theta(z_j|x) p_\theta(x)}{p_\theta(z_j|x)}$.
- $\mathrm{ELBO}(p_\theta(Z|X), \theta) = \sum_{j=1}^{k} p_\theta(z_j|x) \log p_\theta(x) = \log p_\theta(x)$.

Expectation Maximization (EM) algorithm

- Iteratively maximize $\mathrm{ELBO}(q, \theta)$ w.r.t. q and θ.

Expectation Maximization (EM) algorithm

- Iteratively maximize $\mathrm{ELBO}(q, \theta)$ w.r.t. q and θ.
- E step: Fix θ and maximize $\mathrm{ELBO}(q, \theta)$ w.r.t. q. That is, set $q = p_\theta(Z|X)$.

Expectation Maximization (EM) algorithm

- Iteratively maximize $\text{ELBO}(q, \theta)$ w.r.t. q and θ.
- E step: Fix θ and maximize $\text{ELBO}(q, \theta)$ w.r.t. q. That is, set $q = p_\theta(Z|X)$.
- M step: Fix q and maximize $\text{ELBO}(q, \theta)$ w.r.t. θ.

EM algorithm

- E step: $q_{t+1}(Z|X) = p_{\theta_t}(Z|X)$.

EM algorithm

- E step: $q_{t+1}(Z|X) = p_{\theta_t}(Z|X)$.
- M step: $\theta_{t+1} = \mathrm{argmax}_\theta \ \mathsf{ELBO}(q_{t+1}, \theta)$.

EM algorithm

- E step: $q_{t+1}(Z|X) = p_{\theta_t}(Z|X)$.
- M step: $\theta_{t+1} = \mathrm{argmax}_\theta \; \text{ELBO}(q_{t+1}, \theta)$.
- Repeat until convergence.

EM algorithm properties

- EM algorithm decreases $\hat{R}_S(\theta) = \sum_{i=1}^{m} \ell(x_i, \theta)$ at each iteration.
- $\ell(x, \theta_{t+1}) = \mathrm{ELBO}(q_{t+2}, \theta_{t+1}) \geqslant \mathrm{ELBO}(q_{t+1}, \theta_{t+1}) \geqslant \mathrm{ELBO}(q_{t+1}, \theta_t) = \ell(x, \theta_t)$.

Example: mixture of Gaussians

- $x_i \sim \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$. Take $\Sigma_i = I$ for simplicity.

Example: mixture of Gaussians

- $x_i \sim \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$. Take $\Sigma_i = I$ for simplicity.
- That is, $\theta = (\pi_1, \ldots, \pi_k, \mu_1, \ldots, \mu_k)$.

Example: mixture of Gaussians

- $x_i \sim \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$. Take $\Sigma_i = I$ for simplicity.
- That is, $\theta = (\pi_1, \ldots, \pi_k, \mu_1, \ldots, \mu_k)$.
- $p_\theta(x|z_j) = \mathcal{N}(\mu_j, \Sigma_j)$; $p_\theta(z_j) = \pi_j$.

Example: mixture of Gaussians

- $x_i \sim \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$. Take $\Sigma_i = I$ for simplicity.
- That is, $\theta = (\pi_1, \ldots, \pi_k, \mu_1, \ldots, \mu_k)$.
- $p_\theta(x|z_j) = \mathcal{N}(\mu_j, \Sigma_j)$; $p_\theta(z_j) = \pi_j$.
- E step: $q_{t+1}(z|x) = p_{\theta_t}(z|x) = \frac{p_{\theta_t}(x|z)p_{\theta_t}(z)}{p_{\theta_t}(x)}$.

Example: mixture of Gaussians

- $x_i \sim \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$. Take $\Sigma_i = I$ for simplicity.
- That is, $\theta = (\pi_1, \ldots, \pi_k, \mu_1, \ldots, \mu_k)$.
- $p_\theta(x|z_j) = \mathcal{N}(\mu_j, \Sigma_j)$; $p_\theta(z_j) = \pi_j$.
- E step: $q_{t+1}(z|x) = p_{\theta_t}(z|x) = \frac{p_{\theta_t}(x|z) p_{\theta_t}(z)}{p_{\theta_t}(x)}$.
- M step: $\theta_{t+1} = \operatorname{argmax}_\theta \sum_{i=1}^{m} \sum_{j=1}^{k} q_{t+1}(z_j|x_i) \log \frac{p_\theta(x_i, z_j)}{q_{t+1}(z_j|x_i)}$.

Example: mixture of Gaussians

- $x_i \sim \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$. Take $\Sigma_i = I$ for simplicity.
- That is, $\theta = (\pi_1, \ldots, \pi_k, \mu_1, \ldots, \mu_k)$.
- $p_\theta(x|z_j) = \mathcal{N}(\mu_j, \Sigma_j)$; $p_\theta(z_j) = \pi_j$.
- E step: $q_{t+1}(z|x) = p_{\theta_t}(z|x) = \frac{p_{\theta_t}(x|z) p_{\theta_t}(z)}{p_{\theta_t}(x)}$.
- M step: $\theta_{t+1} = \mathrm{argmax}_\theta \sum_{i=1}^{m} \sum_{j=1}^{k} q_{t+1}(z_j|x_i) \log \frac{p_\theta(x_i, z_j)}{q_{t+1}(z_j|x_i)}$.
- $\mu_{j,t+1} = \sum_{i=1}^{m} q_{t+1}(z_j|x_i) x_i$.

Example: mixture of Gaussians

- $x_i \sim \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$. Take $\Sigma_i = I$ for simplicity.
- That is, $\theta = (\pi_1, \ldots, \pi_k, \mu_1, \ldots, \mu_k)$.
- $p_\theta(x|z_j) = \mathcal{N}(\mu_j, \Sigma_j)$; $p_\theta(z_j) = \pi_j$.
- E step: $q_{t+1}(z|x) = p_{\theta_t}(z|x) = \frac{p_{\theta_t}(x|z)p_{\theta_t}(z)}{p_{\theta_t}(x)}$.
- M step: $\theta_{t+1} = \mathrm{argmax}_\theta \sum_{i=1}^{m} \sum_{j=1}^{k} q_{t+1}(z_j|x_i) \log \frac{p_\theta(x_i, z_j)}{q_{t+1}(z_j|x_i)}$.
- $\mu_{j,t+1} = \sum_{i=1}^{m} q_{t+1}(z_j|x_i) x_i$.
- $\pi_{j,t+1} = \frac{1}{m} \sum_{i=1}^{m} q_{t+1}(z_j|x_i) / Z_{j,t+1}$.

VAE revisited

- Hard to compute q_t in general. VAE solves this problem differently.
- $p_\theta(x|z) = \mathcal{N}(f_\theta(z), \sigma^2 I)$. (Decoder: f_θ)
- $q_\phi(z|x) = \mathcal{N}(\mu_\phi(x), \Sigma_\phi(x))$. (Encoder: μ_ϕ, Σ_ϕ)