Clustering and Mixture Models

Ziping Zhao

School of Information Science and Technology
ShanghaiTech University, Shanghai, China

CS182: Introduction to Machine Learning (Fall 2022)
http://cs182.sist.shanghaitech.edu.cn

Ch. 7 of I2ML (Sec. 7.7 excluded)

Outline

Introduction

k-Means Clustering Algorithm

Expectation-Maximization Algorithm

Use of Clustering

Hierarchical Clustering

Choosing the Number of Clusters

Outline

Introduction

k-Means Clustering Algorithm

Expectation-Maximization Algorithm

Use of Clustering

Hierarchical Clustering

Choosing the Number of Clusters

Different Approaches to Density Estimation

- Parametric:
  - $p(\mathbf{x} \mid C_i)$ is represented by a single parametric model.
  - Topic 3 (Parameter Estimation for Generative Models)
- Semiparametric:
  - $p(\mathbf{x} \mid C_i)$ is represented by a mixture of densities.
  - This Topic
- Nonparametric:
  - $p(\mathbf{x} \mid C_i)$ cannot be represented by a single parametric model or a mixture model; the data speaks for itself.
  - Topic 11 (Nonparametric Methods)

- The model flexibility increases (and hence the model bias decreases) from parametric to semiparametric to nonparametric methods.

Mixture Densities

- Mixture (density) model:

$$p(\mathbf{x}) = \sum_{i=1}^{k} p(\mathbf{x} \mid \mathcal{G}_i) P(\mathcal{G}_i)$$

where

$$\begin{aligned} \mathcal{G}_i &= \text{mixture components (or clusters or groups)} \\ p(\mathbf{x} \mid \mathcal{G}_i) &= \text{component densities} \\ P(\mathcal{G}_i) &= \text{mixture proportions (or priors)} \quad \text{(s.t. } P(\mathcal{G}_i) \geq 0, \sum_i P(\mathcal{G}_i) = 1) \end{aligned}$$

The number of components, k, is a hyperparameter to be specified beforehand.

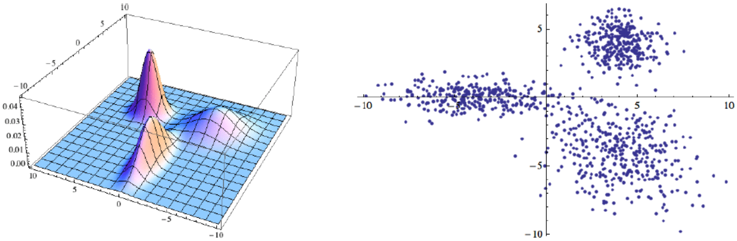
- Given a sample $\mathcal{X}$ and k, learning is to estimate the $p(\mathbf{x} \mid \mathcal{G}_i)$ and $P(\mathcal{G}_i)$.
- Mixture models are also frequently used in statistics and signal processing.

Gaussian Mixture Model

- When we assume that the component densities $p(\mathbf{x} \mid \mathcal{G}_i)$ obey a parametric model, we need only estimate their parameters.
- Gaussian mixture model (GMM) (or mixture of Gaussians):

$$p(\mathbf{x} \mid \mathcal{G}_i) = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

$$\text{parameters } \boldsymbol{\Phi} = \{P(\mathcal{G}_i), \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^{k}$$



(a) A probability distribution on $\mathbb{R}^2$.

(b) Data sampled from this distribution.

Classes vs. Clusters

Supervised	Unsupervised
Sample $\mathcal{X} = \{(\mathbf{x}^{(\ell)}, \mathbf{r}^{(\ell)})\}_{\ell=1}^{N}$	Sample $\mathcal{X} = \{(\mathbf{x}^{(\ell)})\}_{\ell=1}^{N}$
Classes $C_i, i = 1, \ldots, K$ $p(\mathbf{x}) = \sum_{i=1}^{K} p(\mathbf{x} \mid C_i) P(C_i)$ where $p(\mathbf{x} \mid C_i) = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$	Clusters $\mathcal{G}_i, i = 1, \ldots, k$ $p(\mathbf{x}) = \sum_{i=1}^{k} p(\mathbf{x} \mid \mathcal{G}_i) P(\mathcal{G}_i)$ where $p(\mathbf{x} \mid \mathcal{G}_i) = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$
Parameters $\boldsymbol{\Phi} = \{P(C_i), \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^{K}$ $\hat{P}(C_i) = \frac{\sum_\ell r_i^{(\ell)}}{N}$ $\mathbf{m}_i = \frac{\sum_\ell r_i^{(\ell)} \mathbf{x}_i^{(\ell)}}{\sum_\ell r_i^{(\ell)}}$ $\mathbf{S}_i = \frac{\sum_\ell r_i^{(\ell)} (\mathbf{x}_i^{(\ell)} - \mathbf{m}_i)(\mathbf{x}_i^{(\ell)} - \mathbf{m}_i)^T}{\sum_\ell r_i^{(\ell)}}$	Parameters $\boldsymbol{\Phi} = \{P(\mathcal{G}_i), \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^{k}$ (mixture density estimation)

Outline

Introduction

k-Means Clustering Algorithm

Expectation-Maximization Algorithm

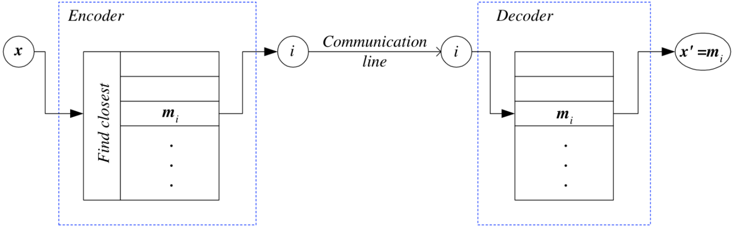
Use of Clustering

Hierarchical Clustering

Choosing the Number of Clusters

k-Means Clustering Algorithm

- Example of clustering problem: color quantization from high to lower resolution (special case of vector quantization (VQ)).
- Problem formulation:
  - Given a sample $\mathcal{X} = \{(\mathbf{x}^{(\ell)})\}_{\ell=1}^{N}$.
  - Find k reference vectors (or prototypes or codebook vectors or codewords) $\mathbf{m}_j$ which best represent the data.
- Encoding/decoding view:
  - Encoding: from a data point $\mathbf{x}^{(\ell)}$ to the index i of a reference vector.
  - Decoding: from an index i to the corresponding reference vector $\mathbf{m}_i$.



Encoding/Decoding

- Each data point $\mathbf{x}^{(\ell)}$ is represented by the index i of the nearest reference vector:
$$i = \arg\min_j \|\mathbf{x}^{(\ell)} - \mathbf{m}_j\|$$
- Encoding can lead to data compression: instead of storing (or transmitting) $\mathbf{x}^{(\ell)}$, we only need to store (or transmit) i.
- Since $\mathbf{x}^{(\ell)}$ is represented by $\mathbf{m}_i$ after encoding and then decoding, reconstruction error $\|\mathbf{x}^{(\ell)} - \mathbf{m}_i\|^2$ is incurred.
- Total reconstruction error:
$$E(\{\mathbf{m}_i\}_{i=1}^k \mid \mathcal{X}) = \sum_\ell \sum_i b_i^{(\ell)} \|\mathbf{x}^{(\ell)} - \mathbf{m}_i\|^2$$

  where
$$b_i^{(\ell)} = \begin{cases} 1 & \text{if } i = \arg\min_j \|\mathbf{x}^{(\ell)} - \mathbf{m}_j\| \\ 0 & \text{otherwise} \end{cases}$$

Optimization Problem

- The best reference vectors are those that minimize the total reconstruction error, so this corresponds to an optimization problem.

$$\begin{aligned} \underset{\{\mathbf{m}_i\}_{i=1}^k, \{b_i\}_{i=1}^k}{\text{minimize}} \quad & \sum_{\ell} \sum_{i} b_i^{(\ell)} \|\mathbf{x}^{(\ell)} - \mathbf{m}_i\|^2 \\ \text{subject to} \quad & b_i^{(\ell)} = \begin{cases} 1 & \text{if } i = \arg\min_j \|\mathbf{x}^{(\ell)} - \mathbf{m}_j\| \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

- However, since $b_i^{(\ell)}$ also depends on $\mathbf{m}_i$, the optimization problem cannot be solved analytically, but iteratively.
- The k-means clustering algorithm is an iterative algorithm for solving the optimization problem.
- The k-means clustering originated from signal processing, where it is also called the Linde-Buzo-Gray (LBG) algorithm.

Algorithm

Initialize $\mathbf{m}_i$, $i = 1, \ldots, k$ (e.g., k randomly selected $\mathbf{x}^{(\ell)}$)

Repeat

For all $\mathbf{x}^{(\ell)} \in \mathcal{X}$, we obtain the estimated labels

$$b_i^{(\ell)} = \begin{cases} 1 & \text{if } i = \arg\min_j \|\mathbf{x}^{(\ell)} - \mathbf{m}_j\| \\ 0 & \text{otherwise} \end{cases}$$

For all $\mathbf{m}_i$, $i = 1, \ldots, k$, we obtain (by taking the derivative of $E(\{\mathbf{m}_i\}_{i=1}^k \mid \mathcal{X})$ with respect to $\mathbf{m}_i$ and setting it to $\mathbf{0}$)

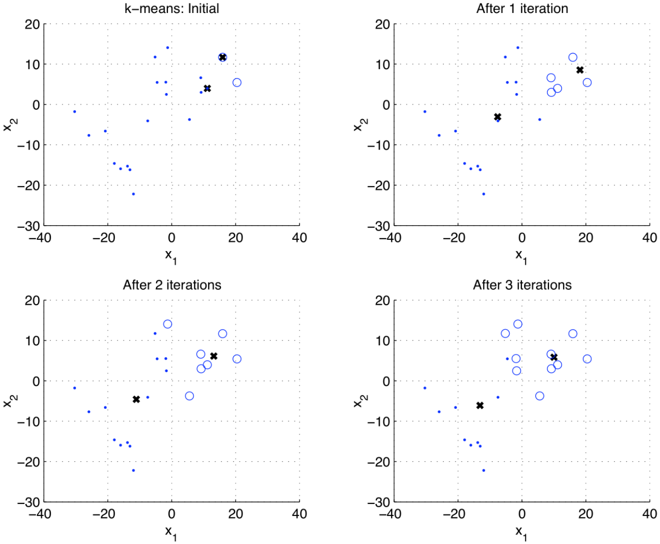
$$\mathbf{m}_i = \frac{\sum_\ell b_i^{(\ell)} \mathbf{x}^{(\ell)}}{\sum_\ell b_i^{(\ell)}}$$

The reference vector is set to the mean (center) of all the instances that it represents.

Until $\mathbf{m}_i$ converge.

- Note $\mathbf{m}_i$ here is the same as the formula for the mean estimation in classification, except that we place the estimated labels $b_i^{(\ell)}$ in place of the labels $r_i^{(\ell)}$.

Evolution of *k*-Means


k−means: Initial

After 1 iteration

After 2 iterations

After 3 iterations

Remarks on k-Means

- The k-means algorithm will converge in finite number of iterations.
- One disadvantage of k-means is that this is a local search procedure, and the final $\mathbf{m}_i$ highly depend on the initial $\mathbf{m}_i$.
- After convergence, all the centers should cover some subset of the data instances and be useful; therefore, it is best to initialize centers where we believe there is data.
- There are also algorithms for adding new centers incrementally or deleting empty ones.
- The k is a hyperparameter to be tuned in the k-means algorithm.
- Vector quantization is one application of clustering, but clustering is also used for preprocessing before a later stage of classification or regression.

Outline

Introduction

k-Means Clustering Algorithm

Expectation-Maximization Algorithm

Use of Clustering

Hierarchical Clustering

Choosing the Number of Clusters

Expectation-Maximization Algorithm

- The expectation-maximization (EM) algorithm may be regarded as a probabilistic extension of k-means, although historically it was not derived as such.
- EM belongs to the general family of minorization-maximization (MM) algorithms.
- EM finds the component density parameters that maximize the likelihood with respect to a mixture model.
- Using a mixture model and given the sample $\mathcal{X} = \{(\mathbf{x}^{(\ell)})\}_{\ell=1}^{N}$, the log likelihood:

$$\mathcal{L}(\mathbf{\Phi} \mid \mathcal{X}) = \log \prod_{\ell} p(\mathbf{x}^{(\ell)} \mid \mathbf{\Phi}) = \sum_{\ell} \log p(\mathbf{x}^{(\ell)} \mid \mathbf{\Phi}) = \sum_{\ell} \log \sum_{i=1}^{k} p(\mathbf{x}^{(\ell)} \mid \mathcal{G}_i) P(\mathcal{G}_i)$$

  where the parameters $\mathbf{\Phi}$ include the priors $P\left(\mathcal{G}_i\right)$ and the sufficient statistics $\mathbf{\Theta}_i$ of the component densities $p(\mathbf{x}^{(\ell)} \mid \mathcal{G}_i)$ (hence, $p(\mathbf{x}^{(\ell)} \mid \mathcal{G}_i) = p(\mathbf{x}^{(\ell)} \mid \mathbf{\Theta}_i)$).
- Optimization of $\mathcal{L}(\mathbf{\Phi} \mid \mathcal{X})$ w.r.t. $\mathbf{\Phi}$ cannot be solved analytically, which calls for an iterative solving procedure.

Iterative Optimization I

- EM is an iterative algorithm for solving the maximum likelihood estimation (MLE) problem.
- EM is suitable for problems that involve two sets of random variables:
  - Observable variables / observed data $\mathcal{X}$
  - Hidden variables / latent data $\mathcal{Z}$
- The $\mathcal{X}$ is also called incomplete data
- The $\{\mathcal{X}, \mathcal{Z}\}$ is called complete data
- There are two reasons why observed data might be incomplete:
  - It's really incomplete: some or all of the instances really have missing values.
  - It's artificially incomplete: it simplifies the math to assume there is extra hidden data.

Iterative Optimization II

- **Goal**: find parameters $\mathbf{\Phi}$ that maximize the likelihood $\mathcal{L}(\mathbf{\Phi} \mid \mathcal{X})$ given sample $\mathcal{X}$, i.e.,

$$\underset{\mathbf{\Phi}}{\text{maximize}} \quad \mathcal{L}(\mathbf{\Phi} \mid \mathcal{X})$$

- EM is suitable if it is difficult to maximize the incomplete-data likelihood $\mathcal{L}(\mathbf{\Phi} \mid \mathcal{X})$ directly but it is easier to work with the complete-data likelihood $\mathcal{L}_C(\mathbf{\Phi} \mid \mathcal{X}, \mathcal{Z})$.
- Because the $\mathcal{Z}$ values are not observed, we cannot work directly with $\mathcal{L}_C(\mathbf{\Phi} \mid \mathcal{X}, \mathcal{Z})$. Instead, we work with an auxiliary function (i.e., the $\mathcal{Q}$ function) which is the expectation of the complete-data likelihood given $\mathcal{X}$ and the current (iteration t) parameter values $\mathbf{\Phi}^t$:

$$\mathcal{Q}(\mathbf{\Phi} \mid \mathbf{\Phi}^t) = E[\mathcal{L}_C(\mathbf{\Phi} \mid \mathcal{X}, \mathcal{Z}) \mid \mathcal{X}, \mathbf{\Phi}^t]$$

E-Step and M-Step

- Given initial point $\mathbf{\Phi}^0$, EM iterates between two steps:
  - E(xpectation)-step: evaluation of expectation (i.e., "computing" the $\mathcal{Q}$ function)

    $$\mathcal{Q}(\mathbf{\Phi} \mid \mathbf{\Phi}^t) = E[\mathcal{L}_C(\mathbf{\Phi} \mid \mathcal{X}, \mathcal{Z}) \mid \mathcal{X}, \mathbf{\Phi}^t]$$

    expectation is taken with respect to the distribution of $\mathcal{Z}$ given $\mathcal{X}$ and $\mathbf{\Phi}^t$.
  - M(aximization)-step: maximization of expectation

    $$\mathbf{\Phi}^{t+1} = \arg\max_{\mathbf{\Phi}} \mathcal{Q}(\mathbf{\Phi} \mid \mathbf{\Phi}^t)$$

- k-means, which is more efficient, may be used to initialize EM.
- An increase in $\mathcal{Q}$ implies an increase in the incomplete-data likelihood:

  $$\mathcal{L}(\mathbf{\Phi}^{t+1} \mid \mathcal{X}) \geq \mathcal{L}(\mathbf{\Phi}^t \mid \mathcal{X})$$

- EM finds a local maximum of the likelihood.

Convergence Proof of EM Algorithm I

- We have

$$
\begin{aligned}
\mathcal{L}(\mathbf{\Phi} \mid \mathcal{X}) = \log p(\mathcal{X} \mid \mathbf{\Phi}) &= \log E_{\mathcal{Z}}[p(\mathcal{X} \mid \mathcal{Z}, \mathbf{\Phi}) \mid \mathbf{\Phi}] \quad \text{(alt. notation: } E_{\mathcal{Z}|\mathbf{\Phi}}[p(\mathcal{X} \mid \mathcal{Z}, \mathbf{\Phi})]\text{)} \\
&= \log E_{\mathcal{Z}}\left[\frac{p(\mathcal{X} \mid \mathcal{Z}, \mathbf{\Phi})}{p(\mathcal{Z} \mid \mathcal{X}, \mathbf{\Phi}^t)} p(\mathcal{Z} \mid \mathcal{X}, \mathbf{\Phi}^t) \mid \mathbf{\Phi}\right] \\
&= \log E_{\mathcal{Z}}\left[\frac{p(\mathcal{X} \mid \mathcal{Z}, \mathbf{\Phi})}{p(\mathcal{Z} \mid \mathcal{X}, \mathbf{\Phi}^t)} p(\mathcal{Z} \mid \mathbf{\Phi}) \mid \mathcal{X}, \mathbf{\Phi}^t\right] \\
&= \log E_{\mathcal{Z}}\left[\frac{p(\mathcal{X}, \mathcal{Z} \mid \mathbf{\Phi})}{p(\mathcal{Z} \mid \mathcal{X}, \mathbf{\Phi}^t)} \mid \mathcal{X}, \mathbf{\Phi}^t\right] \\
\text{(Jensen's Inequality)} \quad &\geq E_{\mathcal{Z}}\left[\log \frac{p(\mathcal{X}, \mathcal{Z} \mid \mathbf{\Phi})}{p(\mathcal{Z} \mid \mathcal{X}, \mathbf{\Phi}^t)} \mid \mathcal{X}, \mathbf{\Phi}^t\right] \\
&= \underbrace{E_{\mathcal{Z}}[\log p(\mathcal{X}, \mathcal{Z} \mid \mathbf{\Phi}) \mid \mathcal{X}, \mathbf{\Phi}^t]}_{= E_{\mathcal{Z}}[\mathcal{L}_C(\mathbf{\Phi}|\mathcal{X},\mathcal{Z})|\mathcal{X},\mathbf{\Phi}^t] = \mathcal{Q}(\mathbf{\Phi}|\mathbf{\Phi}^t)} - E_{\mathcal{Z}}[\log p(\mathcal{Z} \mid \mathcal{X}, \mathbf{\Phi}^t) \mid \mathcal{X}, \mathbf{\Phi}^t] \\
&= \mathcal{B}(\mathbf{\Phi} \mid \mathbf{\Phi}^t)
\end{aligned}
$$

Convergence Proof of EM Algorithm II

- The $\mathcal{B}(\mathbf{\Phi} \mid \mathbf{\Phi}^t)$ function which is called an auxiliary function is a lowerbound surrogate function for $\mathcal{L}(\mathbf{\Phi} \mid \mathcal{X})$ at $\mathbf{\Phi}^t$, i.e.,

$$\mathcal{B}(\mathbf{\Phi} \mid \mathbf{\Phi}^t) \leq \mathcal{L}(\mathbf{\Phi} \mid \mathcal{X}), \quad \forall \mathbf{\Phi}, \mathbf{\Phi}^t$$

- Since

$$\begin{aligned}
\mathcal{B}(\mathbf{\Phi} \mid \mathbf{\Phi}^t) &= E_{\mathcal{Z}}[\log p(\mathcal{X}, \mathcal{Z} \mid \mathbf{\Phi}) \mid \mathcal{X}, \mathbf{\Phi}^t] - E_{\mathcal{Z}}[\log p(\mathcal{Z} \mid \mathcal{X}, \mathbf{\Phi}^t) \mid \mathcal{X}, \mathbf{\Phi}^t] \\
&= E_{\mathcal{Z}}[\log p(\mathcal{X}, \mathcal{Z} \mid \mathbf{\Phi}) \mid \mathcal{X}, \mathbf{\Phi}^t] - E_{\mathcal{Z}}[\log \frac{p(\mathcal{X}, \mathcal{Z} \mid \mathbf{\Phi}^t)}{p(\mathcal{X}, \mid \mathbf{\Phi}^t)} \mid \mathcal{X}, \mathbf{\Phi}^t] \\
&= E_{\mathcal{Z}}[\log p(\mathcal{X} \mid \mathbf{\Phi}^t) \mid \mathcal{X}, \mathbf{\Phi}^t] + E_{\mathcal{Z}}[\log p(\mathcal{X}, \mathcal{Z} \mid \mathbf{\Phi}) \mid \mathcal{X}, \mathbf{\Phi}^t] \\
&\qquad - E_{\mathcal{Z}}[\log p(\mathcal{X}, \mathcal{Z} \mid \mathbf{\Phi}^t) \mid \mathcal{X}, \mathbf{\Phi}^t] \\
&= \underbrace{\log p(\mathcal{X} \mid \mathbf{\Phi}^t)}_{\mathcal{L}(\mathbf{\Phi}^t \mid \mathcal{X})} + E_{\mathcal{Z}}[\log \frac{p(\mathcal{X}, \mathcal{Z} \mid \mathbf{\Phi})}{p(\mathcal{X}, \mathcal{Z} \mid \mathbf{\Phi}^t)} \mid \mathcal{X}, \mathbf{\Phi}^t]
\end{aligned}$$

  we can obtain

$$\mathcal{B}(\mathbf{\Phi}^t \mid \mathbf{\Phi}^t) = \mathcal{L}(\mathbf{\Phi}^t \mid \mathcal{X}), \quad \forall \mathbf{\Phi}^t$$

Convergence Proof of EM Algorithm III

- We have
$$\mathcal{B}(\mathbf{\Phi} \mid \mathbf{\Phi}^t) \leq \mathcal{L}(\mathbf{\Phi} \mid \mathcal{X}), \quad \forall \mathbf{\Phi}, \mathbf{\Phi}^t$$
$$\mathcal{B}(\mathbf{\Phi}^t \mid \mathbf{\Phi}^t) = \mathcal{L}(\mathbf{\Phi}^t \mid \mathcal{X}), \quad \forall \mathbf{\Phi}^t$$
indicating $\mathcal{B}(\mathbf{\Phi} \mid \mathbf{\Phi}^t)$ is a tight lowerbound function for $\mathcal{L}(\mathbf{\Phi} \mid \mathcal{X})$ at $\mathbf{\Phi}^t$.
- Since
$$\begin{aligned}\mathbf{\Phi}^{t+1} &= \arg\max_{\mathbf{\Phi}} \mathcal{Q}(\mathbf{\Phi} \mid \mathbf{\Phi}^t) \\ &= \arg\max_{\mathbf{\Phi}} \mathcal{Q}(\mathbf{\Phi} \mid \mathbf{\Phi}^t) - E_{\mathcal{Z}}[\log p(\mathcal{Z} \mid \mathcal{X}, \mathbf{\Phi}^t) \mid \mathcal{X}, \mathbf{\Phi}^t] \\ &= \arg\max_{\mathbf{\Phi}} \mathcal{B}(\mathbf{\Phi} \mid \mathbf{\Phi}^t)\end{aligned}$$
we have
$$\mathcal{L}(\mathbf{\Phi}^{t+1} \mid \mathcal{X}) \geq \mathcal{B}(\mathbf{\Phi}^{t+1} \mid \mathbf{\Phi}^t) \geq \mathcal{B}(\mathbf{\Phi}^t \mid \mathbf{\Phi}^t) = \mathcal{L}(\mathbf{\Phi}^t \mid \mathcal{X})$$

From EM Algorithm to MM

- The $\mathcal{Q}$ function evaluated in EM is called a special case of the minorizing functions under the hood of minorization-maximization (MM) algorithmic framework.
- A minorizing function $\underline{\mathcal{L}}$, a tight lowerbound of $\mathcal{L}$, can be chosen in a more general way (not only relying on the Jensen's inequality) as long as satisfying

$$\begin{aligned} \underline{\mathcal{L}}(\boldsymbol{\Phi} \mid \boldsymbol{\Phi}^t) &\leq \mathcal{L}(\boldsymbol{\Phi} \mid \mathcal{X}), \quad \forall \boldsymbol{\Phi}, \boldsymbol{\Phi}^t \\ \underline{\mathcal{L}}(\boldsymbol{\Phi}^t \mid \boldsymbol{\Phi}^t) &= \mathcal{L}(\boldsymbol{\Phi}^t \mid \mathcal{X}), \quad \forall \boldsymbol{\Phi}^t \end{aligned}$$

  which can lead to "cheap" (say, analytical) update for the following maximization steps.

EM Algorithm for Mixture Model I

- In the case of mixture models, hidden variables are the sources of observations, namely, which observation belongs to which component.
- Indicator variables $\mathbf{z}^{(\ell)} = (z_1^{(\ell)}, \ldots, z_k^{(\ell)})^T$, i.e. the labels for $\mathbf{x}^{(\ell)}$:

$$z_i^{(\ell)} = \begin{cases} 1 & \text{if } \mathbf{x}^{(\ell)} \text{ belongs to cluster } \mathcal{G}_i \\ 0 & \text{otherwise} \end{cases}$$

- EM algorithm for mixture model
  - E-step: estimating the labels $\mathbf{z}^{(\ell)}$ of the observations given our current knowledge of the components
  - M-step: updating the component knowledge $\boldsymbol{\Phi}$ given the labels estimated
- These two steps in EM are the same as the two steps of k-means.
- Let the prior probabilities $P(\mathcal{G}_i) = \pi_i$ for brevity, so

$$P(\mathbf{z}^{(\ell)}) = \prod_{i=1}^{k} \pi_i^{z_i^{(\ell)}}$$

EM Algorithm for Mixture Model II

- Likelihood of an observed variable $\mathbf{x}^{(\ell)}$ given hidden variable $\mathbf{z}^{(\ell)}$ is equal to its probability specified by the component that generated it:

$$p(\mathbf{x}^{(\ell)} \mid \mathbf{z}^{(\ell)}) = \prod_{i=1}^{k} p_i(\mathbf{x}^{(\ell)})^{z_i^{(\ell)}}$$

  where $p_i(\mathbf{x}^{(\ell)})$ is the shorthand for $p(\mathbf{x}^{(\ell)} \mid \mathcal{G}_i)$.
- Joint density (or likelihood of the variable $(\mathbf{x}^{(\ell)}, \mathbf{z}^{(\ell)})$):

$$p(\mathbf{x}^{(\ell)}, \mathbf{z}^{(\ell)}) = P(\mathbf{z}^{(\ell)}) p(\mathbf{x}^{(\ell)} \mid \mathbf{z}^{(\ell)})$$

- Complete-data log likelihood given the i.i.d. sample $\mathcal{X}$:

$$\begin{aligned}\mathcal{L}_C(\mathbf{\Phi} \mid \mathcal{X}, \mathcal{Z}) &= \log \prod_{\ell} p(\mathbf{x}^{(\ell)}, \mathbf{z}^{(\ell)} \mid \mathbf{\Phi}) = \sum_{\ell} \log p(\mathbf{x}^{(\ell)}, \mathbf{z}^{(\ell)} \mid \mathbf{\Phi}) \\ &= \sum_{\ell} \left[\log P(\mathbf{z}^{(\ell)} \mid \mathbf{\Phi}) + \log p(\mathbf{x}^{(\ell)} \mid \mathbf{z}^{(\ell)}, \mathbf{\Phi}) \right] \\ &= \sum_{\ell} \sum_{i} z_i^{(\ell)} \left[\log \pi_i + \log p_i(\mathbf{x}^{(\ell)} \mid \mathbf{\Phi}) \right]\end{aligned}$$

E-Step I

- Evaluation of $\mathcal{Q}(\mathbf{\Phi} \mid \mathbf{\Phi}^t)$:

$$
\begin{aligned}
\mathcal{Q}(\mathbf{\Phi} \mid \mathbf{\Phi}^t) =& E[\mathcal{L}_C(\mathbf{\Phi} \mid \mathcal{X}, \mathcal{Z}) \mid \mathcal{X}, \mathbf{\Phi}^t] \\
=& \sum_{\ell} \sum_{i} E[z_i^{(\ell)} \mid \mathcal{X}, \mathbf{\Phi}^t] \left[\log \pi_i + \log p_i(\mathbf{x}^{(\ell)} \mid \mathbf{\Phi})\right]
\end{aligned}
$$

where

$$
\begin{aligned}
E[z_i^{(\ell)} \mid \mathcal{X}, \mathbf{\Phi}^t] =& E[z_i^{(\ell)} \mid \mathbf{x}^{(\ell)}, \mathbf{\Phi}^t] = P\left(z_i^{(\ell)} = 1 \mid \mathbf{x}^{(\ell)}, \mathbf{\Phi}^t\right) \\
=& \frac{p\left(\mathbf{x}^{(\ell)} \mid z_i^{(\ell)} = 1, \mathbf{\Phi}^t\right) P\left(z_i^{(\ell)} = 1 \mid \mathbf{\Phi}^t\right)}{p\left(\mathbf{x}^{(\ell)} \mid \mathbf{\Phi}^t\right)} = \frac{p_i\left(\mathbf{x}^{(\ell)} \mid \mathbf{\Phi}^t\right) \pi_i}{\sum_j p_j\left(\mathbf{x}^{(\ell)} \mid \mathbf{\Phi}^t\right) \pi_j} \\
=& \frac{p\left(\mathbf{x}^{(\ell)} \mid \mathcal{G}_i, \mathbf{\Phi}^t\right) P(\mathcal{G}_i)}{\sum_j p\left(\mathbf{x}^{(\ell)} \mid \mathcal{G}_j, \mathbf{\Phi}^t\right) P(\mathcal{G}_j)} = P(\mathcal{G}_i \mid \mathbf{x}^{(\ell)}, \mathbf{\Phi}^t) \\
\equiv& h_i^{(\ell)}
\end{aligned}
$$

E-Step II

- The expected value of the hidden variable, $h_i^{(\ell)}$, is the posterior probability that $\mathbf{x}^{(\ell)}$ is generated by component $\mathcal{G}_i$, i.e., $P(\mathcal{G}_i \mid \mathbf{x}^{(\ell)}, \mathbf{\Phi}^t)$.
- Since the posterior probability $h_i^{(\ell)} \in [0, 1]$, it can be seen as a soft label, as opposed to the hard label ($b_i^{(\ell)} \in \{0, 1\}$) of k-means.
- Strictly speaking the E-step only computes $h_i^{(\ell)}$ but not $\mathcal{Q}(\mathbf{\Phi} \mid \mathbf{\Phi}^t)$.
- Specifically, for Gaussian components, $\hat{p}_i(\mathbf{x}^{(\ell)} \mid \mathbf{\Theta}_i) = \mathcal{N}(\mathbf{m}_i, \mathbf{S}_i)$, and so

$$h_i^{(\ell)} = \frac{|\mathbf{S}_i|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{x}^{(\ell)} - \mathbf{m}_i)^T \mathbf{S}_i^{-1} (\mathbf{x}^{(\ell)} - \mathbf{m}_i)\right] \pi_i}{\sum_j |\mathbf{S}_j|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{x}^{(\ell)} - \mathbf{m}_j)^T \mathbf{S}_j^{-1} (\mathbf{x}^{(\ell)} - \mathbf{m}_j)\right] \pi_j}$$

M-Step I

- Maximization of $\mathcal{Q}(\boldsymbol{\Phi} \mid \boldsymbol{\Phi}^t)$:

$$\begin{aligned}
\boldsymbol{\Phi}^{t+1} &= \arg\max_{\boldsymbol{\Phi}} \mathcal{Q}(\boldsymbol{\Phi} \mid \boldsymbol{\Phi}^t) = \arg\max_{\boldsymbol{\Phi}} \left[\sum_\ell \sum_i h_i^{(\ell)} \left[\log \pi_i + \log p_i(\mathbf{x}^{(\ell)} \mid \boldsymbol{\Phi}) \right] \right] \\
&= \arg\max_{\boldsymbol{\Phi}} \left[\sum_\ell \sum_i h_i^{(\ell)} \log \pi_i + \sum_\ell \sum_i h_i^{(\ell)} \log p_i(\mathbf{x}^{(\ell)} \mid \boldsymbol{\Phi}) \right] \\
&= \arg\max_{\boldsymbol{\Phi}} \left[\sum_\ell \sum_i h_i^{(\ell)} \log \pi_i + \sum_\ell \sum_i h_i^{(\ell)} \log p_i(\mathbf{x}^{(\ell)} \mid \boldsymbol{\Theta}_i) \right]
\end{aligned}$$

Or, equivalently, the problem is

$$\begin{aligned}
\underset{\{\pi_i\},\{\boldsymbol{\Theta}_i\}}{\text{maximize}} \quad & \mathcal{Q}(\boldsymbol{\Phi} \mid \boldsymbol{\Phi}^t) = \sum_\ell \sum_i h_i^{(\ell)} \log \pi_i + \sum_\ell \sum_i h_i^{(\ell)} \log p_i(\mathbf{x}^{(\ell)} \mid \boldsymbol{\Theta}_i) \\
\text{subject to} \quad & \sum_i \pi_i = 1
\end{aligned}$$

M-Step II

- The second term of $\mathcal{Q}(\mathbf{\Phi} \mid \mathbf{\Phi}^t)$ does not depend on π_i. The problem for $\{\pi_i\}$ is

$$\begin{aligned} \underset{\{\pi_i\}}{\text{maximize}} \quad & \sum_\ell \sum_i h_i^{(\ell)} \log \pi_i \\ \text{subject to} \quad & \sum_i \pi_i = 1 \end{aligned}$$

Using the constraint $\sum_i \pi_i = 1$ to define the Lagrangian, we solve for

$$\frac{\partial}{\partial \pi_i} \left[\sum_\ell \sum_i h_i^{(\ell)} \log \pi_i - \lambda (\sum_i \pi_i - 1) \right] = 0$$

to get

$$\pi_i = \frac{\sum_\ell h_i^{(\ell)}}{N}$$

which is analogous to the prior estimation in classification.

M-Step III

- The first term of $\mathcal{Q}(\mathbf{\Phi} \mid \mathbf{\Phi}^t)$ does not depend on $\mathbf{\Theta}_i$. The problem for $\{\mathbf{\Theta}_i\}$ is

$$\underset{\{\mathbf{\Theta}_i\}}{\text{maximize}} \quad \sum_{\ell} \sum_{i} h_i^{(\ell)} \log p_i(\mathbf{x}^{(\ell)} \mid \mathbf{\Theta}_i)$$

  We solve for

$$\frac{\partial}{\partial \mathbf{\Theta}_i} \sum_{\ell} \sum_{i} h_i^{(\ell)} \log p_i \left(\mathbf{x}^{(\ell)} \mid \mathbf{\Phi}\right) = \frac{\partial}{\partial \mathbf{\Theta}_i} \sum_{\ell} h_i^{(\ell)} \log p_i \left(\mathbf{x}^{(\ell)} \mid \mathbf{\Theta}_i\right) = \mathbf{0}$$

  In general, analytical solutions may not be attainable.

- Remark: To attain a closed-form solution for the M(aximization)-step, the general minorization-maximization (MM) algorithmic framework can be applied. The M(inorization)-step in MM is more flexible than the E(xpectation)-step in EM to obtain a lower bound surrogate function for the likelihood objective function.

M-Step IV

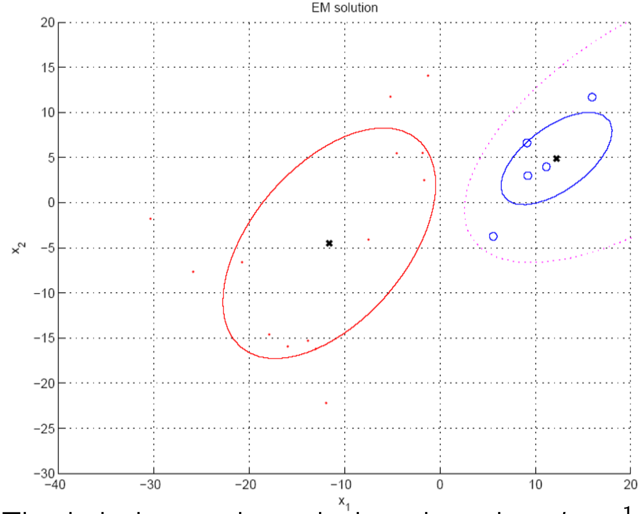
- Specifically, for Gaussian components, $\hat{p}_i(\mathbf{x}^{(\ell)} \mid \boldsymbol{\Theta}_i) = \mathcal{N}(\mathbf{m}_i, \mathbf{S}_i)$, and so we get

$$\mathbf{m}_i^{t+1} = \frac{\sum_\ell h_i^{(\ell)} \mathbf{x}^{(\ell)}}{\sum_\ell h_i^{(\ell)}} \quad \text{and} \quad \mathbf{S}_i^{t+1} = \frac{\sum_\ell h_i^{(\ell)} (\mathbf{x}^{(\ell)} - \mathbf{m}_i^{t+1})(\mathbf{x}^{(\ell)} - \mathbf{m}_i^{t+1})^T}{\sum_\ell h_i^{(\ell)}}$$

  which are analogous to the mean and covariance estimation in classification under Gaussian assumption with the estimated soft labels $h_i^{(\ell)}$ replacing the actual labels $r_i^{(\ell)}$.

- Remark: When $h_i^{(\ell)}$ are used instead of $b_i^{(\ell)}$, an instance contributes to the update of parameters of all components, to each proportional to that probability. This is especially useful if the instance is close to the midpoint between two centers.

Example



The dashed curve shows the boundary where $h_i = \frac{1}{2}$.

Regularization in GMM

- Just as in parametric classification, with small samples and large dimensionality we can regularize by making simplifying assumptions.
- When $\hat{p}_i(\mathbf{x}^{(\ell)} \mid \mathbf{\Theta}_i) = \mathcal{N}(\mathbf{m}_i, \mathbf{S})$ (with a shared covariance matrix)

$$\underset{\{\mathbf{m}_i\},\mathbf{S}}{\text{minimize}} \quad \frac{Nk}{2} \log|\mathbf{S}| + \frac{1}{2}\sum_{\ell}\sum_{i} h_i^{(\ell)} (\mathbf{x}^{(\ell)} - \mathbf{m}_i)^T \mathbf{S}^{-1} (\mathbf{x}^{(\ell)} - \mathbf{m}_i)$$

- When $\hat{p}_i(\mathbf{x}^{(\ell)} \mid \mathbf{\Theta}_i) = \mathcal{N}(\mathbf{m}_i, s^2\mathbf{I})$ (with a shared diagonal covariance matrix)

$$\underset{\{\mathbf{m}_i\},s}{\text{minimize}} \quad \frac{Nk}{2} \log\left|s^2\mathbf{I}\right| + \frac{1}{2}\sum_{\ell}\sum_{i} h_i^{(\ell)} \frac{\|\mathbf{x}^{(\ell)} - \mathbf{m}_i\|^2}{s^2}$$

and (with equal priors)

$$h_i^{(\ell)} = \frac{\exp\left[-\frac{1}{2s^2}\|\mathbf{x}^{(\ell)} - \mathbf{m}_i\|^2\right]}{\sum_j \exp\left[-\frac{1}{2s^2}\|\mathbf{x}^{(\ell)} - \mathbf{m}_j\|^2\right]}$$

k-Means Clustering as EM

- The *k*-means clustering is a special case of EM applied to GMM where
  - inputs are assumed independent with equal and shared variances,
  - all components have equal priors,
  - labels are hardened (called hard EM in this case).

Introducing Model Bias

- When the sample is small, overtting may occur.
- Possible solutions via introducing model bias (for Gaussian mixtures):
  - Constraining the covariance matrices of the Gaussian components, e.g., use shared (assuming clusters have the same shape) or diagonal (removing all correlations) covariance matrices.
  - Applying PCA or FA to perform dimensionality reduction (to be covered in future topic) in the components, e.g., mixtures of latent variable models:

    $$p(\mathbf{x}^{(\ell)} \mid \mathcal{G}_i) = \mathcal{N}(\mathbf{m}_i, \mathbf{V}_i \mathbf{V}_i^T + \boldsymbol{\Psi}_i)$$

    which realizes clustering and dimensionality reduction simultaneously.

Outline

Introduction

k-Means Clustering Algorithm

Expectation-Maximization Algorithm

Use of Clustering

Hierarchical Clustering

Choosing the Number of Clusters

Data Exploration

- Like dimensionality reduction methods (to be discussed later), clustering can be used for data exploration:
  - Dimensionality reduction methods: find correlations between features (and thus group features).
  - Clustering methods: find similarities between instances (and thus group instances).
- Clustering allows knowledge extraction through:
  - Number of clusters
  - Prior probabilities
  - Cluster parameters

Clustering as Preprocessing

- The dimensionality reduction methods (from $\mathbf{x}^{(\ell)} \in \mathbb{R}^d$ to $\mathbf{z}^{(\ell)} \in \mathbb{R}^k$) allowed us to make a mapping to a new space.
- After clustering, the estimated group labels ($h_i^{(\ell)}$ for soft labels and $b_i^{(\ell)}$ for hard labels) may be seen as the dimensions of a new k-dimensional space.
- Local vs. distributed representation:
  - Clustering gives local representation: only one $b_i^{(\ell)}$ is 1 and all others are 0; or only a few $h_i^{(\ell)}$ are nonzero.
  - Dimensionality reduction gives distributed representation: in general all z_i are nonzero.
- Supervised learning tasks such as classication and regression can be performed in the new space.

  Advantage: the preprocessing stage does not need labeled data.

Mixture of Mixtures in Classification

- In classification, when each class is a mixture model composed of a number of components, the whole density is a mixture of mixtures:

$$p(\mathbf{x} \mid C_i) = \sum_{j=1}^{k_i} p(\mathbf{x} \mid \mathcal{G}_{ij}) P(\mathcal{G}_{ij})$$

$$p(\mathbf{x}) = \sum_{i=1}^{K} p(\mathbf{x} \mid C_i) P(C_i)$$

  where k_i is the number of components making up $p(\mathbf{x} \mid C_i)$ and $\mathcal{G}_{ij}$ is the component j of class i. Note that different classes may need different number of components.

- Learning the parameters of components is done separately for each class.

Outline

Introduction

k-Means Clustering Algorithm

Expectation-Maximization Algorithm

Use of Clustering

Hierarchical Clustering

Choosing the Number of Clusters

Hierarchical Clustering

- We have discussed clustering from a probabilistic point of view as fitting a mixture model to the data, i.e., maximizing the log likelihood, or in terms of finding codewords minimizing reconstruction error.
- Hierarchical clustering generally refers to methods that cluster instances into a hierarchical structure (called dendrogram) based solely on some similarity or distance measure between them.

Distance Measures

- Distance (or similarity) measures defined between instances:
  - Minkowski distance:

$$d_m(\mathbf{x}^{(r)}, \mathbf{x}^{(s)}) = \Big(\sum_{j=1}^{d} |x_j^{(r)} - x_j^{(s)}|^p\Big)^{1/p}$$

  - Euclidean distance: special case of Minkowski distance with $p = 2$
  - Manhattan (or city-block) distance: special case of Minkowski distance with $p = 1$

$$d_{cb}(\mathbf{x}^{(r)}, \mathbf{x}^{(s)}) = \sum_{j=1}^{d} |x_j^{(r)} - x_j^{(s)}|$$

  - Hamming distance:

$$d_h(\mathbf{x}^{(r)}, \mathbf{x}^{(s)}) = \sum_{j=1}^{d} \mathbf{1}_{[x_j^{(r)} \neq x_j^{(s)}]}$$

Distance between Groups $\mathcal{G}_i$ and $\mathcal{G}_j$

- Single-link clustering:

$$d_{min}(\mathcal{G}_i, \mathcal{G}_j) = \min_{\mathbf{x}^{(r)} \in \mathcal{G}_i, \mathbf{x}^{(s)} \in \mathcal{G}_j} d(\mathbf{x}^{(r)}, \mathbf{x}^{(s)})$$

- Complete-link clustering:

$$d_{max}(\mathcal{G}_i, \mathcal{G}_j) = \max_{\mathbf{x}^{(r)} \in \mathcal{G}_i, \mathbf{x}^{(s)} \in \mathcal{G}_j} d(\mathbf{x}^{(r)}, \mathbf{x}^{(s)})$$

- Average-link clustering:

$$d_{avg}(\mathcal{G}_i, \mathcal{G}_j) = \frac{1}{|\mathcal{G}_i| \cdot |\mathcal{G}_j|} \sum_{\mathbf{x}^{(r)} \in \mathcal{G}_i, \mathbf{x}^{(s)} \in \mathcal{G}_j} d(\mathbf{x}^{(r)}, \mathbf{x}^{(s)})$$

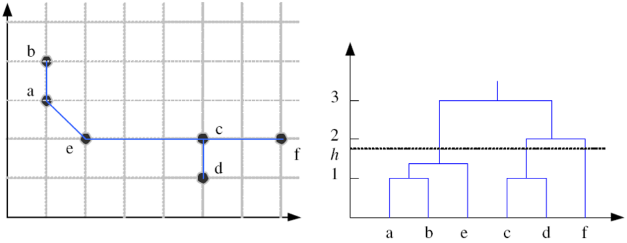
- Other possibilities, e.g., distance between group means (centroids).

Agglomerative vs. Divisive Clustering

- An agglomerative clustering algorithm starts with N groups each with one instance.
  - At each iteration, the two most similar groups are merged, reducing the number of groups by one.
  - The process stops when there is only one group left.
- A divisive clustering algorithm starts with one group and goes in the opposite direction.
  - At each iteration, large groups are divided into smaller groups, increasing the number of groups by one.
  - The process stops when each group contains a single instance.
- The process can be intersected at any level to get the desired number of clusters.

Single-Link Method for Agglomerative Clustering

- Consider a weighted, completely connected graph with nodes corresponding to instances and edges between nodes with weights equal to the distances between the instances.
- The single-link method is to constructing the minimal spanning tree of this graph.
- The agglomerative clustering procedure can be drawn as a hierarchical structure called dendrogram (can be intersected at any level to get the clusters):



- Single-link and complete-link methods calculate the distance between groups differently that affect the clusters and the dendrogram.

Outline

Introduction

k-Means Clustering Algorithm

Expectation-Maximization Algorithm

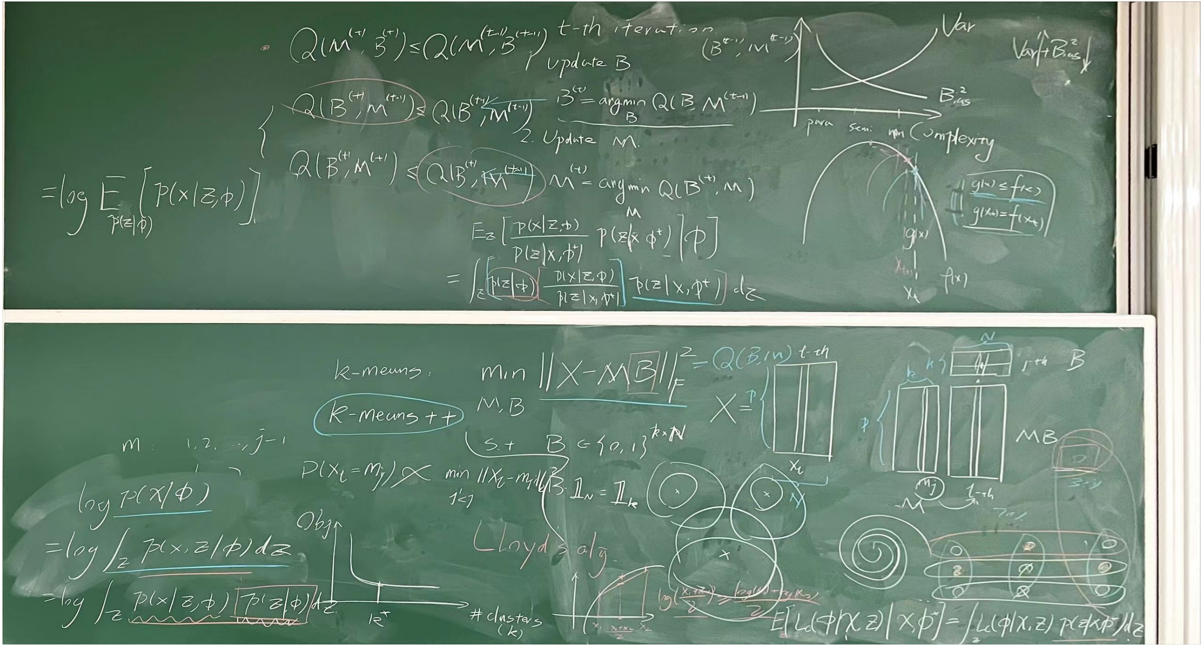
Use of Clustering

Hierarchical Clustering

Choosing the Number of Clusters

Choosing k

- This is a model selection issue for clustering.
- Some common possibilities:
  - Defined by the application, e.g., image quantization.
  - Visualize the data (e.g., in 2-D using dimensionality reduction methods) and check for the number of clusters.
  - Incremental approach: add one at a time and monitor the reconstruction error, log likelihood, or intergroup distances.
  - Domain experts check the clustering result.
- One method to mitigate the effect of k is using the Bayesian estimation for mixture models.



$Q(M^{(t)}, B^{(t)}) \le Q(M^{(t-1)}, B^{(t-1)})$ t-th iteration $(B^{(t-1)}, M^{(t-1)})$

1. update B

$Q(B^{(t)}, M^{(t-1)}) \le Q(B^{(t-1)}, M^{(t-1)})$ $B^{(t)} = \underset{B}{\arg\min}\, Q(B, M^{(t-1)})$

2. Update M.

$Q(B^{(t)}, M^{(t)}) \le Q(B^{(t)}, M^{(t-1)})$ $M^{(t)} = \underset{M}{\arg\min}\, Q(B^{(t)}, M)$

$= \log E_{p(z|\phi)}\left[p(x|z,\phi)\right]$

$E_z\left[\frac{p(x|z,\phi)}{p(z|x,\phi^t)} p(z|x\,\phi^t) \middle| \phi\right]$

$= \int_z \left[p(z|\phi) \frac{p(x|z,\phi)}{p(z|x,\phi^t)}\right] p(z|x,\phi^t)\, dz$

k-means: $\underset{M,B}{\min} \|X - MB\|_F^2 = Q(B, M)$ t-th

K-means ++

s.t. $B \in \{0,1\}^{k \times N}$

$B \cdot \mathbb{1}_N = \mathbb{1}_k$

$m: 1, 2, \dots, j-1$

$P(x_l = m_j) \propto \underset{j'<j}{\min} \|x_l - m_{j'}\|^2$

Lloyd's alg.

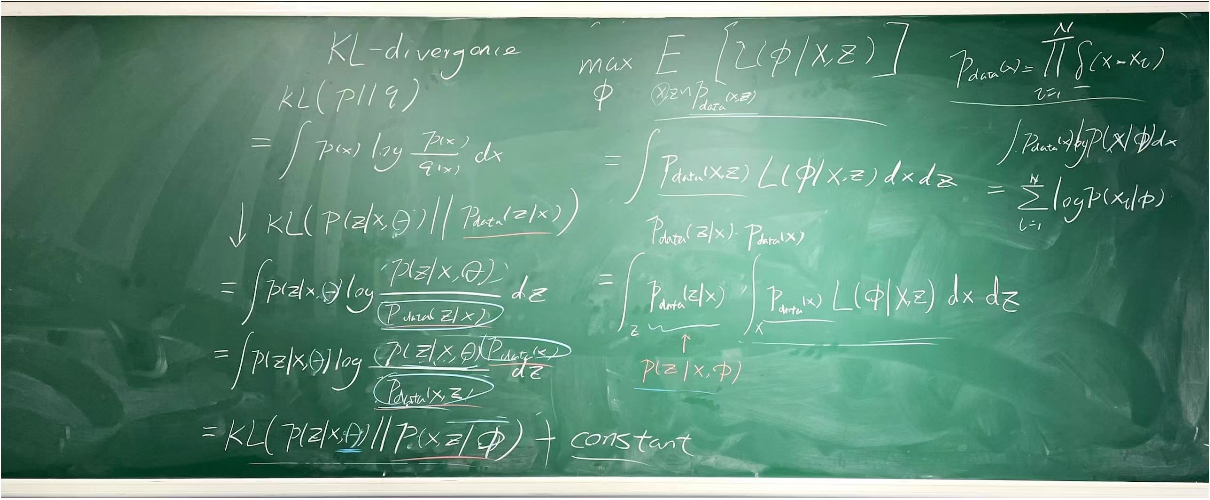
$\log p(X|\phi)$

$= \log \int_z p(x, z|\phi)\, dz$

$= \log \int_z p(x|z,\phi)\, p(z|\phi)\, dz$

$\log\left(\frac{x_1+x_2}{2}\right) \ge \frac{\log(x_1) + \log(x_2)}{2}$

$E\left[L_c(\phi|X,Z) \middle| X, \phi^t\right] = \int_z L_c(\phi|X,Z)\, p(z|x,\phi^t)\, dz$



KL-divergonce

$$KL(p||q)$$

$$= \int p(x) \log \frac{p(x)}{q(x)} dx$$

$$\downarrow \; KL(p(z|x,\theta) \,||\, p_{data}(z|x))$$

$$= \int p(z|x,\theta) \log \frac{p(z|x,\theta)}{p_{data}(z|x)} dz$$

$$= \int p(z|x,\theta) \log \frac{p(z|x,\theta)\, p_{data}(x)}{p_{data}(x,z)} dz$$

$$= KL(p(z|x,\theta) \,||\, p(x,z|\Phi)) + \text{constant}$$

$$\max_{\phi} \; E_{x,z \sim p_{data}(x,z)} \left[L(\phi|x,z) \right]$$

$$= \int p_{data}(x,z)\, L(\phi|x,z)\, dx\, dz$$

$$p_{data}(z|x) \cdot p_{data}(x)$$

$$= \int_z \underset{\overset{\uparrow}{p(z|x,\phi)}}{p_{data}(z|x)} \int_x p_{data}(x)\, L(\phi|x,z)\, dx\, dz$$

$$p_{data}(x) = \prod_{i=1}^{N} \delta(x - x_i)$$

$$\int p_{data}(x) \log p(x|\phi)\, dx$$

$$= \sum_{i=1}^{N} \log p(x_i|\phi)$$