Clustering and Expectation Maximization

Shan-Hung Wu
shwu@cs.nthu.edu.tw

Department of Computer Science,
National Tsing Hua University, Taiwan

NetDB-ML, Spring 2015

Outline

Outline

Clustering

- We now consider the unsupervised datasets $\mathcal{X} = \{\boldsymbol{x}^{(t)}\}_{t=1}^{N}$ where labels $\boldsymbol{r}^{(t)}$ are missing
  - Learning the a posteriori knowledge from unlabeled data is called the ***unsupervised learning***
- ***Clustering*** is one unsupervised learning technique used to identify the groups $G_1, \cdots, G_K$ in each which instances are similar (or close) to each other
  - K could be either predefined (a hyperparameter) or not (a parameter)
- Output: $\mathcal{Z} := \{\boldsymbol{z}^{(t)}\}_t$, where
  - Hard labeling: $\boldsymbol{z}^{(t)} \in \{0,1\}^K$ and $z_i^{(t)} = 1$ iff the instance t belongs to group i
  - Soft labeling: $\boldsymbol{z}^{(t)} \in \mathbb{R}^K$ and $z_i^{(t)}$ denotes the degree (e.g., probability) the instance t belongs to group i

Applications

- Pattern recognition: groups may be meaningful
  - E.g., product/user cluster in market analysis
- Compression: instances in the same group can be represented by a prototype
- Data labeling: groups are good hints for labels
- Data reprocessing for classification/regression: attributes of instances can be augmented by group information; or we can identify groups in each class to estimate $P[\boldsymbol{x}|C_i]$ and $P[C_i]$ more precisely
- And so on...

Clustering vs. Dimensionality Reduction

- In dimensionality reduction, we find correlations between ***attributes*** and "group" (i.e., select/extract) attributes
- In clustering, we find similarities between ***instances*** and group instances

Outline

K-Means Clustering (1/2)

- Suppose each group G_i is parametrized by a prototype $\boldsymbol{m}_i$, the mean of all instances in this group
- Hard labeling: $z_i^{(t)} = 1$ iff $\boldsymbol{x}^{(t)}$ is the closest to $\boldsymbol{m}_i$; i.e., $\left\|\boldsymbol{x}^{(t)} - \boldsymbol{m}_i\right\| = \min_j \left\|\boldsymbol{x}^{(t)} - \boldsymbol{m}_j\right\|$
- The objective of ***K-means clustering*** is to find $\boldsymbol{m}_i$ such that the total ***reconstruction error*** $rec(\{\boldsymbol{m}_i\}_{i=1}^K; \mathcal{X}) = \sum_{t=1}^N \sum_{i=1}^K z_i^{(t)} \left\|\boldsymbol{x}^{(t)} - \boldsymbol{m}_i\right\|^2$ is minimized

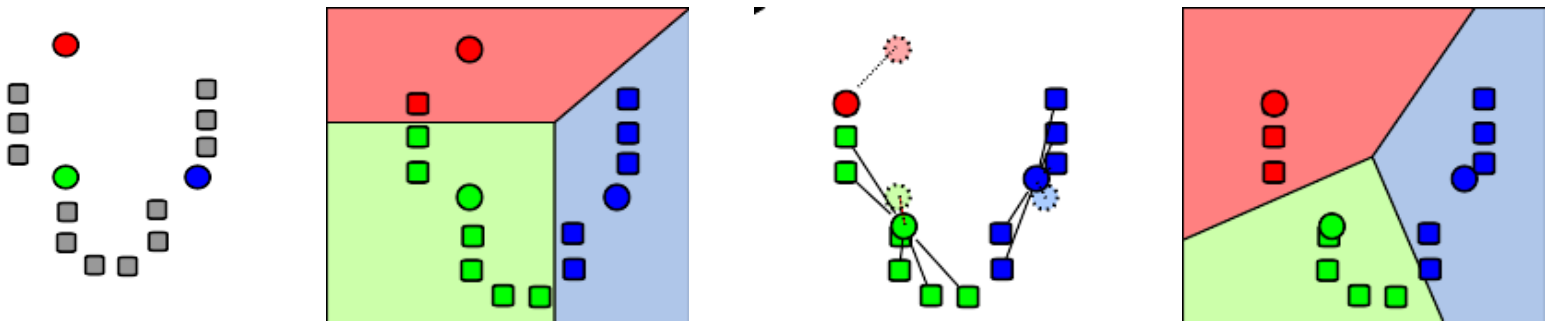
K-Means Clustering (2/2)

Input: $\mathcal{X} \leftarrow \{\boldsymbol{x}^{(t)}\}_{t=1}^{N}$, K
Output: The prototypes $\boldsymbol{m}_i$, $1 \leqslant i \leqslant K$

Initialize each $\boldsymbol{m}_i$ to a random example $\boldsymbol{x}^{(t)}$;
repeat
  foreach $\boldsymbol{x}^{(t)} \in \mathcal{X}$ **do**
    $z_i^{(t)} \leftarrow \begin{cases} 1 & \text{if } \left\|\boldsymbol{x}^{(t)} - \boldsymbol{m}_i\right\| = \min_j \left\|\boldsymbol{x}^{(t)} - \boldsymbol{m}_j\right\| \\ 0 & \text{otherwise} \end{cases}$;
  end
  foreach $\boldsymbol{m}_i$ **do**
    $\boldsymbol{m}_i \leftarrow \sum_{t=1}^{N} z_i^{(t)} \boldsymbol{x}^{(t)} / \sum_{t=1}^{N} z_i^{(t)}$;
  end
until *all* $\boldsymbol{m}_i$ *converge*;

Algorithm 1: The K-means algorithm.

Example



Applications (1/2)

- One famous application of the K-means clustering is ***vector quantization***, which aims to find a discrete set of vectors $\{\boldsymbol{m}_i\}_{i=1}^K$ representative of the whole, possibly continuous, set of data points
  - E.g., in ***color quantization***, we seeks the best 256 colors of an 24 bits/pixel (16 million) color image
  - Once we get these 256 colors, for each pixel we only need to store the 8 bits color index
- We can quantize the 16 million colors uniformly into 256, but some of these 256 colors may be wasted when there is no nearby color appears in the image
  - We want nonuniform quantization where $\boldsymbol{m}_i$ sit at the most dense areas of the whole dataset
- The K-means clustering minimizes $rec(\{\boldsymbol{m}_i\}_{i=1}^K; \mathcal{X}) = \sum_{t=1}^N \sum_{i=1}^K z_i^{(t)} \left\|\boldsymbol{x}^{(t)} - \boldsymbol{m}_i\right\|^2$ and finds prototypes at the center of the dense regions

Applications (2/2)

- Another example is the use of ***codebooks*** in telecommunication systems
  - Each point in the dataset is a vector storing the sample of a voice signal
  - We want to quantize samples into K representative vectors
  - If we store these K vectors in each device, the signal can be sent by indexes (of $\lg K$ bits each) only

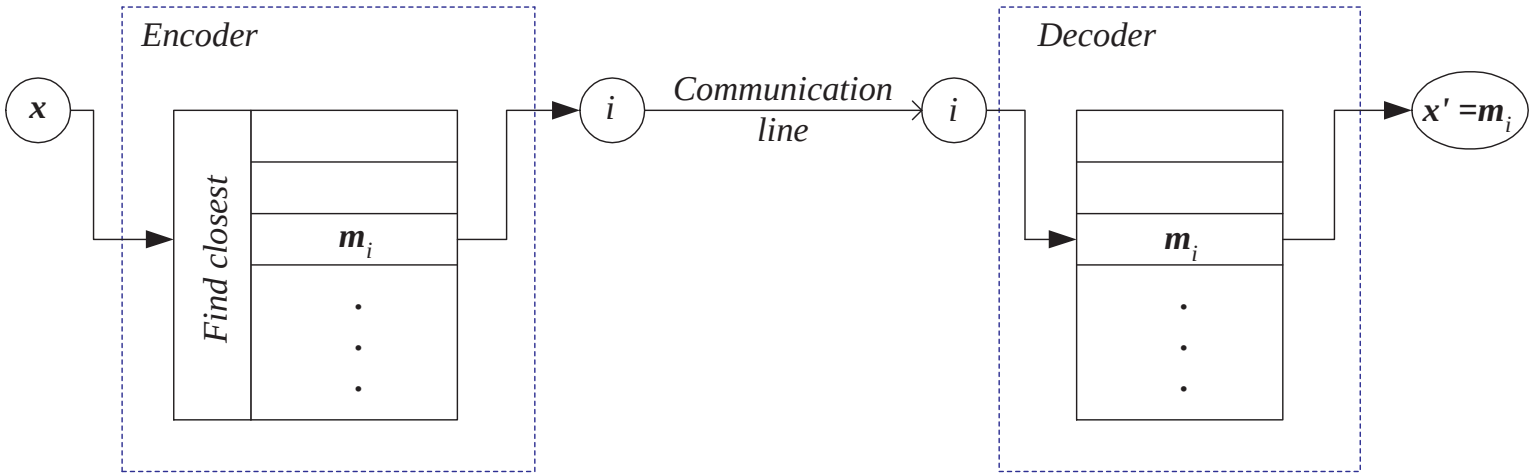


Figure : Given $\boldsymbol{x}$, the encoder sends the index i of the nearest ***codeword*** $\boldsymbol{m}_i$ and the decoder receives $\boldsymbol{x}' = \boldsymbol{m}_i$. The error is $\|\boldsymbol{x} - \boldsymbol{x}'\|^2$.

Limitations (1/2)

- The main disadvantage of the K-means clustering is that it is a local search procedure
  - The final prototypes $\boldsymbol{m}_i$ may not be the optimal ones, and highly depend on the initial $\boldsymbol{m}_i$
  - Can you give an example dataset based on which the K-means returns bad clusters? [Homework]
- Generally, the initial $\boldsymbol{m}_i$ should a) locate at regions where instances occur; b) be far away from each other
- The K-means++ proposes one possible initialization step:
  1. Choose an instance uniformly at random to be $\boldsymbol{m}_1$
  2. For each $\boldsymbol{x}^{(t)}$, compute $d(\boldsymbol{x}^{(t)})$, the distance between $\boldsymbol{x}^{(t)}$ and the nearest $\boldsymbol{m}_i$ that has already been determined
  3. Assign another instance to $\boldsymbol{m}_{i+1}$, but this time an instance $\boldsymbol{x}$ is chosen with probability $\frac{d(\boldsymbol{x})^2}{\sum_{t=1}^{N} d(\boldsymbol{x}^{(t)})^2}$
  4. Repeat Steps 2 and 3 until K initial prototypes are determined

Limitations (2/2)

- Another shortcoming of the K-means is that clusters are assumed to be spherical and with equal size
  - Due to that the Euclidean distance is used when updating the cluster assignment $z_i^{(t)}$ for each instance
- In practice, clusters may have different sizes
  - Next, we see how the above assumption can be relaxed using the probability framework we are already familiar

Outline

Mixture Models

- Basic assumption: the dataset $\mathcal{X}$ is a mixture of groups $G_1, \cdots, G_K$
  - E.g., in the hand-written digit recognition, $\mathcal{X}$ consists of images of "0," "1," "2," and so forth
  - Even if $\mathcal{X}$ are images of the same digit (say "1") there are still typical different ways to write the digit (with or without head)
- Soft labeling: $\mathcal{Z} = \{\boldsymbol{z}^{(t)} \in \mathbb{R}^K\}_t$
- The ***mixture density*** of an instance $\boldsymbol{x}$ can be expressed as $p(\boldsymbol{x}) = \sum_{i=1}^{K} p(\boldsymbol{x}|G_i)P[G_i]$
- Model: a collection of groups, i.e., $\{G_i\}_{i=1}^{K}$
- Parameters: G_1, $\cdots$, and G_K
- Objective: $\arg_{G_1,\cdots,G_K} \max \prod_{t=1}^{N} \sum_{i=1}^{K} p(\boldsymbol{x}^{(t)}|G_i)P[G_i]d\boldsymbol{x}$
- A generative model this case

Parametric vs. Nonparametric vs. Semiparametric

- Parametric models: models that can be completely described by (a small number of) parameters
- Nonparametric models: those that cannot be described by parameters
- Semiparametric models: those that can be partially described by parameters
  - Each cluster is parametric
  - But the mixture of clusters, $\mathcal{Z} = \{\boldsymbol{z}^{(t)}\}_t$, is not (i.e., we do not assume the mixture to follow some distribution)

Semiparametric Clustering vs. Parametric Classification

- Parametric classification is a special case of mixture model where the groups (i.e., classes) are known in advance:
  $P[\boldsymbol{x}^{(t)}] = \sum_{i=1}^{K} p(\boldsymbol{x}^{(t)}|C_i)P[C_i]d\boldsymbol{x}$
- Assume $p(\boldsymbol{x}^{(t)}|C_i)d\boldsymbol{x}$ and $P[C_i]$ follow Gaussian and Bernoulli distributions parametrized by $\theta_i = (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ and $\theta_i' = p_i$ respectively
- Since we know which instance belongs to which class by $\boldsymbol{r}^{(t)}$, we can estimate θ_i and θ_i' analytically by maximizing $P[\mathcal{X}|\theta_i]$ and $P[\mathcal{X}|\theta_i']$:
  - $\widehat{p_i} = \frac{N_i}{N}$, where $N_i = \sum_{t=1}^{N} r_i^{(t)}$
  - $\boldsymbol{m}_i = \frac{1}{N_i} \sum_{t=1}^{N} \boldsymbol{x}^{(t)} r_i^{(t)}$ and $\boldsymbol{S}_i = \frac{1}{N_i - 1} \sum_{t=1}^{N} r_i^{(t)} (\boldsymbol{x}^{(t)} - \boldsymbol{m}_i)(\boldsymbol{x}^{(t)} - \boldsymbol{m}_i)^\top$
- Unfortunately, in semiparametric clustering we don't know $\boldsymbol{z}^{(t)}/\boldsymbol{r}^{(t)}$ so we cannot solve $p(\boldsymbol{x}^{(t)}|C_i)d\boldsymbol{x}$ and $P[C_i]$ analytically

If No $\{\theta_i, \theta_i'\}_{i=1}^K$, Make Them Up

- How?

If No $\{\theta_i, \theta_i'\}_{i=1}^K$, Make Them Up

- How? Borrowing the iterations from K-means
- Start from a random guess of $\{\theta_i, \theta_i'\}_{i=1}^K$ and then perform the following two steps iteratively:
  1. For each instance $\boldsymbol{x}^{(t)}$, update its $\boldsymbol{z}^{(t)}$ based on the current $G_1, \cdots, G_K$ parametrized by $\theta_1, \cdots, \theta_K$
  2. Update $\theta_1, \cdots, \theta_K$ based on the current $\boldsymbol{z}^{(t)}$
- Stop until the groups do not change in Step 2 (or the changes of groups are smaller than a threshold ε)

Semiparametric Density Estimation (1/2)

- Suppose in the mixture density $p(\boldsymbol{x}) = \sum_{i=1}^{K} p(\boldsymbol{x}|G_i)P[G_i]$, each $p(\boldsymbol{x}^{(t)}|G_i)$ and $P[G_i]$ are Gaussian and Bernoulli distributions parametrized by $\theta_i = (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ and $\theta_i' = \pi_i$ respectively
- Denote the collection of estimators by $\Theta = (\boldsymbol{m}_i, \boldsymbol{S}_i, \pi_i)_{i=1}^{K}$
- We guess initial Θ, and then:
  1. Update mixture: $z_i^{(t)} = P[z_i^{(t)}|\boldsymbol{x}^{(t)}; \Theta] = \frac{p(\boldsymbol{x}^{(t)}|z_i^{(t)}; \Theta)P[z_i^{(t)}; \Theta]}{p(\boldsymbol{x}^{(t)}; \Theta)} = \frac{p(\boldsymbol{x}^{(t)}|z_i^{(t)}; \Theta)\pi_i}{\sum_{j=1}^{K} p(\boldsymbol{x}^{(t)}|z_j^{(t)}; \Theta)\pi_j} = \frac{det(\boldsymbol{S}_i)^{-1/2} exp[-(1/2)(\boldsymbol{x}^{(t)} - \boldsymbol{m}_i)^\top \boldsymbol{S}_i^{-1}(\boldsymbol{x}^{(t)} - \boldsymbol{m}_i)]\pi_i}{\sum_{j=1}^{K} det(\boldsymbol{S}_j)^{-1/2} exp[-(1/2)(\boldsymbol{x}^{(t)} - \boldsymbol{m}_j)^\top \boldsymbol{S}_j^{-1}(\boldsymbol{x}^{(t)} - \boldsymbol{m}_j)]\pi_j}$
     - Unlike in K-means, we assign ***soft labels*** to $z_i^{(t)}$
  2. Update Θ: knowing $z_i^{(t)}$, we can update π_i, $\boldsymbol{m}_i$, and $\boldsymbol{S}_i$ by, e.g., maximizing the likelihood $P[\mathcal{X}|\Theta]$

Semiparametric Density Estimation (2/2)

Input: $\mathcal{X} \leftarrow \{\boldsymbol{x}^{(t)}\}_{t=1}^{N}$, K
Output: $\Theta = (\boldsymbol{m}_i, \boldsymbol{S}_i, \pi_i)_{i=1}^{K}$

Initialize Θ by performing several iterations of K-means;
repeat
  foreach $\boldsymbol{x}^{(t)} \in \mathcal{X}$ **do**
    $$z_i^{(t)} \leftarrow \frac{det(\boldsymbol{S}_i)^{-1/2} exp[-(1/2)(\boldsymbol{x}^{(t)} - \boldsymbol{m}_i)^\top \boldsymbol{S}_i^{-1}(\boldsymbol{x}^{(t)} - \boldsymbol{m}_i)]\pi_i}{\sum_{j=1}^{K} det(\boldsymbol{S}_j)^{-1/2} exp[-(1/2)(\boldsymbol{x}^{(t)} - \boldsymbol{m}_j)^\top \boldsymbol{S}_j^{-1}(\boldsymbol{x}^{(t)} - \boldsymbol{m}_j)]\pi_j};$$
  end
  foreach π_i, $\boldsymbol{m}_i$, *and* $\boldsymbol{S}_i$ **do**
    $$\pi_i \leftarrow \frac{\sum_{t=1}^{N} z_i^{(t)}}{N};$$
    $$\boldsymbol{m}_i \leftarrow \frac{\sum_{t=1}^{N} \boldsymbol{x}^{(t)} z_i^{(t)}}{\sum_{t=1}^{N} z_i^{(t)}} \text{ and } \boldsymbol{S}_i \leftarrow \frac{\sum_{t=1}^{N} z_i^{(t)} (\boldsymbol{x}^{(t)} - \boldsymbol{m}_i)(\boldsymbol{x}^{(t)} - \boldsymbol{m}_i)^\top}{\sum_{t=1}^{N} z_i^{(t)}};$$
  end
until Θ *converges*;

Algorithm 2: Semiparametric density estimation for Gaussian mixtures.

Simplifications

- As in parametric classification, with small training dataset and large dimensionality we can regularize our model by simplifying assumptions
- When the priors $P[G_i] = \pi_i$ are all equal and $\boldsymbol{S}_i = s^2\boldsymbol{I}$, we have $z_i^{(t)} = \frac{exp\left[-(1/2s^2)\left\|\boldsymbol{x}^{(t)} - \boldsymbol{m}_i\right\|\right]}{\sum_{j=1}^{K} exp\left[-(1/2s^2)\left\|\boldsymbol{x}^{(t)} - \boldsymbol{m}_j\right\|\right]}$
- We thus see that the K-means clustering is just a special case of the semiparametric density estimation applied to Gaussian mixtures, where
  - Attributes of instances are independent and with equal variance
  - All groups have equal priors
  - Labels are hardened

Outline

Hierarchical Clustering (1/2)

- So far, we assume that clusters are independent groups (although they may overlap)
- In some applications, we may want to find the hierarchy of clusters
- Two common types of algorithms:
  - ***Agglomerative***: Starting from N groups, each with single instance, iteratively merging two most similar groups to form a larger one, until there remains a single group
  - ***Divisive***: Starting one group containing all instances, dividing large groups into smaller ones, until there are N groups

Hierarchical Clustering (2/2)

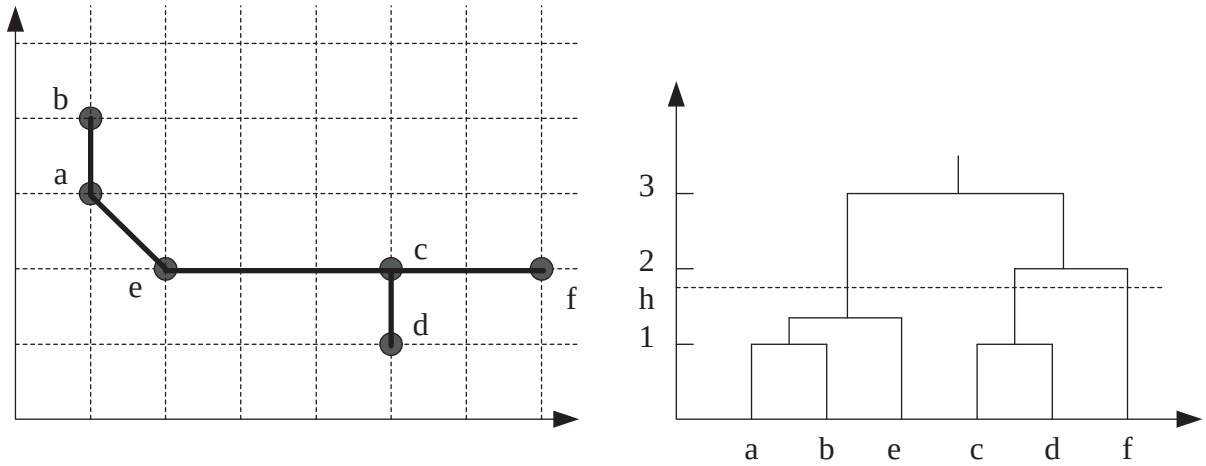
- When deciding which groups should be merged (or split), a measure of similarity, or equivalently distance d, is required
  - One common choice is the Minkowski distance:
    $d(\boldsymbol{x}^{(r)}, \boldsymbol{x}^{(s)}) = \left(\sum_{i=1}^{d} \left| x_i^{(r)} - x_i^{(s)} \right|^p \right)^{1/p}$ for some p
- But how to calculate the distance between two groups?

Hierarchical Clustering (2/2)

- When deciding which groups should be merged (or split), a measure of similarity, or equivalently distance d, is required
  - One common choice is the Minkowski distance:
    $d(\boldsymbol{x}^{(r)}, \boldsymbol{x}^{(s)}) = \left(\sum_{i=1}^{d} \left| x_i^{(r)} - x_i^{(s)} \right|^p \right)^{1/p}$ for some p
- But how to calculate the distance between two groups?
  - ***Single-link*** metric: $d(G_i, G_j) = \min_{\boldsymbol{x}^{(r)} \in G_i, \boldsymbol{x}^{(s)} \in G_j} d(\boldsymbol{x}^{(r)}, \boldsymbol{x}^{(s)})$
  - ***Complete-link*** metric: $d(G_i, G_j) = \max_{\boldsymbol{x}^{(r)} \in G_i, \boldsymbol{x}^{(s)} \in G_j} d(\boldsymbol{x}^{(r)}, \boldsymbol{x}^{(s)})$

Dendrograms

- The result of hierarchical clustering can be shown as the ***dendrogram***:



- Each internal node corresponds to a group
- The height of the internal node denote the distance between groups
- The dendrogram can be intersected at a user-specific level h to get the clusters
- In each cluster, instances in the input space are connected as a tree

Single- or Complete-Link?

Single- or Complete-Link?

- With the complete-link metric, all instance in a group have distance less than h
  - Assumes that each cluster is spherical
  - Similar to k-means and semiparametric density estimation
  - Used only when this assumption is likely to be true
- Single-link clusters may have diameter (i.e., the greatest length of the shortest paths between instances) much larger than h
  - With the single-link metric, two instance are grouped together at level h if
    - The distance between them is less than h; or
    - There exists a path between them such that any two consecutive instances along the path have mutual distance less than h
  - Each final cluster may have an arbitrary shape
  - Suitable for clusters backed by respective underlying manifolds

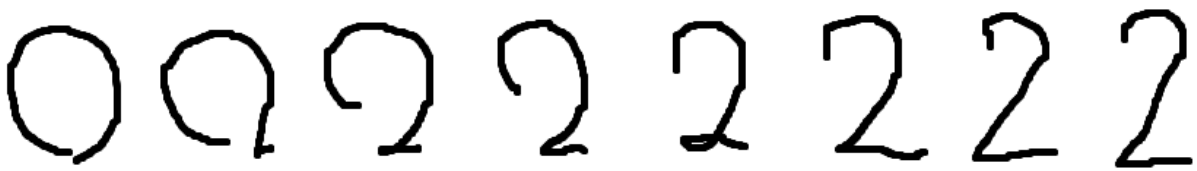
Outline

Global vs. Local Models

- We have seen models that find clusters by assuming some structure for each cluster
- Global structure: each cluster represents a dense region of a known shape
  - E.g., k-means, semiparametric density estimation, hierarchical clustering with complete-link metric
- Local similarity: each instance in a cluster is similar to its nearby instances
  - E.g., hierarchical clustering with single-link metric
  - Local models can produce clusters of arbitrary shapes
  - Suitable to datasets where clusters are backed by respective underlying manifolds

More Local Models

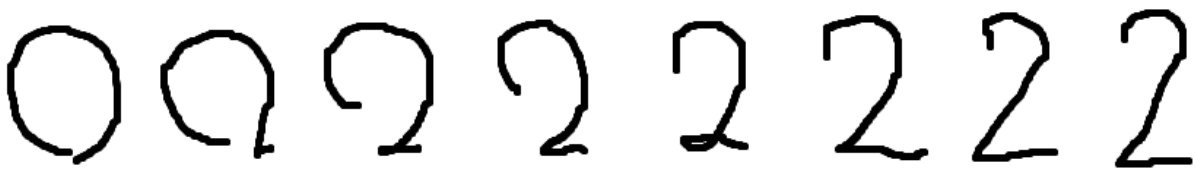
- In local models, any two instances in the same cluster are ***not*** necessarily similar
  - This is both an advantage and disadvantage
- Cons: they tends to find clusters of unbalanced sizes



  - Outliers form singleton clusters
- How to make clusters balanced?

More Local Models

- In local models, any two instances in the same cluster are ***not*** necessarily similar
  - This is both an advantage and disadvantage
- Cons: they tends to find clusters of unbalanced sizes



  - Outliers form singleton clusters
- How to make clusters balanced?
- We consider only the the flat clustering in the next

Balanced Cut of Local Similarity Graph (1)

- Given a set of data points $\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(N)}$. Let $\boldsymbol{S} \in \mathbb{R}^{N \times N}$ be the local similarity matrix where $s_{i,j} \geqslant 0$ is the similarity between instance between instances i and j ***if they are neighbors***
  - Euclidean distance is clearly not a good choice
- Local similarity measure?

Balanced Cut of Local Similarity Graph (1)

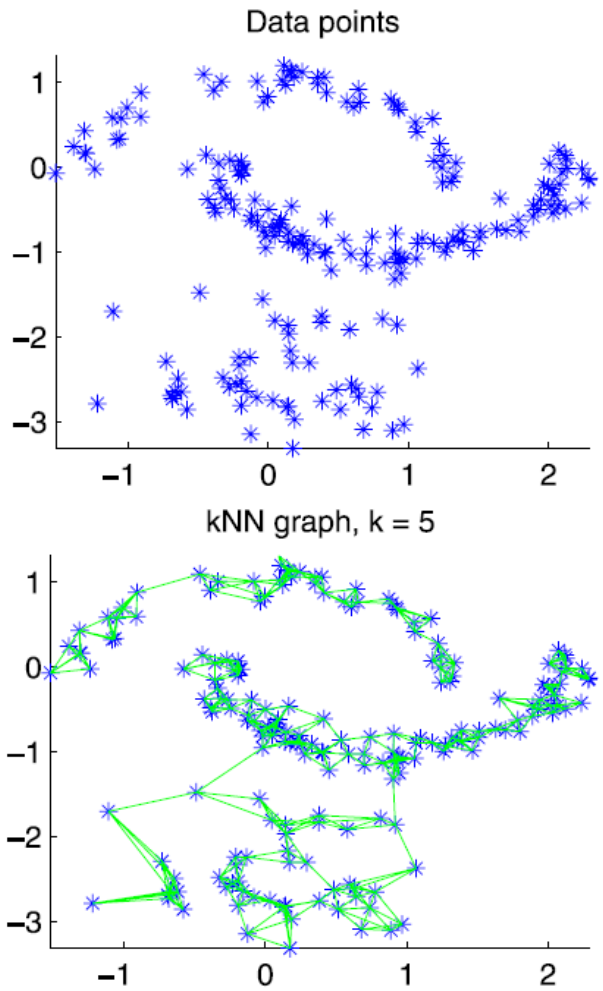
- Given a set of data points $\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(N)}$. Let $\boldsymbol{S} \in \mathbb{R}^{N \times N}$ be the local similarity matrix where $s_{i,j} \geqslant 0$ is the similarity between instance between instances i and j ***if they are neighbors***
  - Euclidean distance is clearly not a good choice
- Local similarity measure?
  - ϵ-NN similarity: $s_{i,j}$ inverse proportional to the Euclidean distance between i and j if i is a ϵ-nearest neighbor of j or vice versa; otherwise 0
  - Gaussian similarity (soft ϵ-NN): $s_{i,j} = \exp(-\frac{\|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\|^2}{\sigma^2})$ for some hyperparameter σ

Balanced Cut of Local Similarity Graph (2)

- Consider the graph $G = (V, E)$ where V denotes the set of instances and E denotes the set of non-zero local similarity scores
- Given a set of nodes $A \subset V$, define $Cut(A) := \sum_{i \in A, j \notin A} s_{i,j}$
- We want to find a k-partition $A_1, \cdots, A_K$ of V that solve the problem:

$$\arg \min_{A_1, \cdots, A_K \subset V} RatioCut(A_1, \cdots, A_K) := \frac{1}{2} \sum_{i=1}^{K} \frac{Cut(A_i)}{|A_i|}$$

  - Cross-partition links are edges are minimized
  - $|A_1|, \cdots, |A_K|$ are balanced



Relaxation

- Unfortunately, although the min-cut problems can be solved efficiently, the balanced min-cut problems are NP-hard
- ***Spectral clustering*** solves a relaxation of the above problem
  - Finds the eigenvectors of a ***graph Laplacian matrix*** induced from the local similarity graph
  - Efficient

Graph Laplacian

- Given a (local) similarity matrix $\boldsymbol{S}$, the ***graph Laplacian matrix*** is defined as

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{S},$$

  where $\boldsymbol{D}$ is an $N \times N$ diagonal matrix with $d_i = \sum_{j=1}^{N} s_{i,j}$ on the diagonal

- For any vector $\boldsymbol{f} \in \mathbb{R}^N$, we have $\boldsymbol{f}^\top \boldsymbol{L} \boldsymbol{f} = \frac{1}{2} \sum_{i,j=1}^{N} s_{i,j} \left(f_i - f_j\right)^2$ [Homework]
- $\boldsymbol{L}$ is symmetric and positive semi-definite
  - The smallest eigenvalue of $\boldsymbol{L}$ is 0, and the constant one vector $\mathbf{1} \in \mathbb{R}^N$ must be (one of) the corresponding eigenvector
  - $\boldsymbol{L}$ has N non-negative eigenvalues $0 = \lambda_1 \leqslant \cdots \leqslant \lambda_N$.

Spectral Clustering

- Idea: map each $\boldsymbol{x}^{(t)} \in \mathbb{R}^N$ to $\boldsymbol{z}^{(t)} \in \mathbb{R}^m$ in some low dimensional space such that $\boldsymbol{z}^{(i)}$ and $\boldsymbol{z}^{(j)}$ are similar if they belong to the same cluster
  - Then apply a traditional clustering algorithm (e.g., k-means) to obtain the final cluster
- Based on $\boldsymbol{f}^\top \boldsymbol{L}\boldsymbol{f} = \frac{1}{2}\sum_{i,j=1}^{N} s_{i,j}\left(f_i - f_j\right)^2$, we can first solve

$$\arg\min_{\boldsymbol{F}=[\boldsymbol{f}^{(1)},\cdots \boldsymbol{f}^{(m)}]\in\mathbb{R}^{N\times m}} \operatorname{tr}(\boldsymbol{F}^\top \boldsymbol{L}\boldsymbol{F}) = \sum_{i=1}^{m} \boldsymbol{f}^{(i)\top}\boldsymbol{L}\boldsymbol{f}^{(i)},$$
$$\text{subject to } \boldsymbol{F}^\top\boldsymbol{F} = \boldsymbol{I}$$

  and then let $\boldsymbol{z}^{(t)}$ be the t-th row of $\boldsymbol{F}$
  - $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ are orthogonal so that they provide complementary perspectives
  - Each $\boldsymbol{f}_i$ is normalized so that the clusters are balanced (to be explained later)

Spectrum of L (1)

- From the Rayleigh-Ritz theorem, $\boldsymbol{f}^{(1)}, \cdots, \boldsymbol{f}^{(m)}$ are the eigenvectors corresponding to the smallest eigenvalues of $\boldsymbol{L}$

Theorem

Let $G = (V, E)$ be an undirected graph with non-negative weights. Then the multiplicity K of the eigenvalue 0 of $\boldsymbol{L}$ equals the number of connected components $A_1, \cdots, A_K \subset V$ in the graph. The eigenspace of eigenvalue 0 is spanned by the indicator vectors $\mathbf{1}_{A_1}, \cdots, \mathbf{1}_{A_K} \in \mathbb{R}^N$ of those components.

Spectrum of L (2)

Proof.

Assume that $\boldsymbol{f}$ is an eigenvector with eigenvalue 0. We know that $0 = \boldsymbol{f}^\top \boldsymbol{L}\boldsymbol{f} = \sum_{i,j=1}^{N} s_{i,j} \left(f_i - f_j\right)^2$. As $s_{i,j}$ is non-negative, the sum can only vanish if all terms vanish. Thus, if two vertices v_i and v_j are connected (i.e., $s_{i,j} > 0$), then $f_i = f_j$. When $K = 1$, $\boldsymbol{f}$ needs to be constant one vector and $\boldsymbol{L}$ has eigenvalue 0 with multiplicity 1. When $K > 1$, without loss of generality we assume that the vertices are ordered according to the connected components they belong to. Then $\boldsymbol{S}$ has a block diagonal form, and the same is true for $\boldsymbol{L}$:

$$\boldsymbol{L} = \begin{pmatrix} \boldsymbol{L}_1 & & \\ & \ddots & \\ & & \boldsymbol{L}_K \end{pmatrix}.$$

Since the spectrum of $\boldsymbol{L}$ is given by the union of the spectra of $\boldsymbol{L}_i$, and the corresponding eigenvectors of $\boldsymbol{L}$ are the eigenvectors of $\boldsymbol{L}_i$, filled with 0 at the positions of the other blocks. □

Spectrum of L (3)

- Based on the above theorem, we should make sure that #connected components $< m$ when constructing the local similarity graph
  - Otherwise, some cluster may contain one connected component, and some may contain multiple
- In practice, we usually construct a fully-connected graph
  - The eigenvector of 0 is $\mathbf{1}$
- Other than $\mathbf{1}$, what $\boldsymbol{f}$ makes $\boldsymbol{f}^\top \boldsymbol{L}\boldsymbol{f} = \frac{1}{2}\sum_{i,j=1}^{N} s_{i,j}\left(f_i - f_j\right)^2$ small?
  - Those $\boldsymbol{f}$'s with value levels
  - Coordinates corresponding to the same G_i have the same value (forming a level)
  - The gap between different levels corresponds to the min-cuts

Spectrum of L (4)

- Besides, $\|\boldsymbol{f}\| = 1$ makes gap correspond to the balanced min-cuts
- For example, suppose $K = 2$. Let

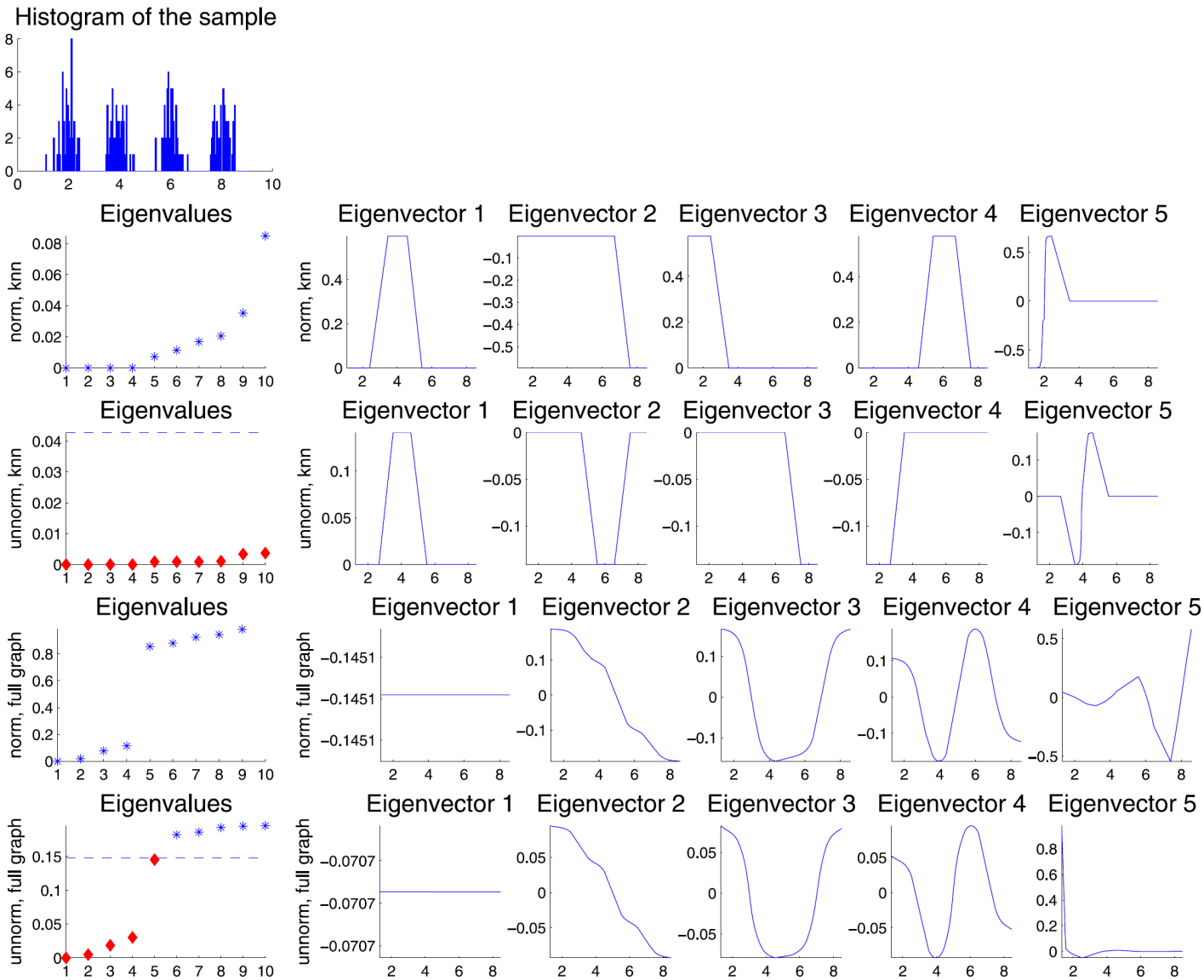
$$f_i = \begin{cases} \sqrt{\frac{|\bar{G}|}{|V||G|}}, & \text{if } \boldsymbol{x}^{(i)} \in \boldsymbol{G} \\ -\sqrt{\frac{|G|}{|V||\bar{G}|}}, & \text{otherwise} \end{cases}.$$

We have

$$tr(\boldsymbol{f}^\top \boldsymbol{L}\boldsymbol{f}) = RatioCut\left(G, \bar{G}\right),$$

furthermore, $\boldsymbol{f}^\top \mathbf{1} = 0$ and $\|\boldsymbol{f}\| = 1$ [Homework]

$m = K$ is enough due to orthogonality



Spectral Clustering Algorithms

Input: Similarity matrix $\boldsymbol{S}$, number of clusters K
Output: Clusters $A_1, \cdots, A_K$

Compute the Laplacian $\boldsymbol{L}$.;
Compute the first K eigenvectors $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_K$ of $\boldsymbol{L}$. Let $\boldsymbol{U} \in \mathbb{R}^{N \times K}$ be the matrix containing the vectors $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_K$ as columns.;
For $i = 1, \cdots, N$, let $\boldsymbol{y}_i \in \mathbb{R}^K$ be the vector corresponding to the i-th row of $\boldsymbol{U}$.;
Cluster the points $(\boldsymbol{y}_i)_{i=1,\cdots,N}$ with the K-means algorithm;

Algorithm 3: The spectral clustering algorithm.

Pros and Cons

- Pros:
  - Local model, balanced
  - Efficient event for large datasets (as $\boldsymbol{S}$ is sparse)
  - No issue of getting stuck in local minimum (e.g., as in k-means due to bad initializations)
- Cons:
  - Performance sensitive to the quality of the local similarity graph
  - Relaxation is loose: no guarantee that the final clusters correspond to the balanced min-cuts
- Which local similarity is better?
  - Empirically, ϵ-NN graph is less vulnerable to the imperfect choice of parameters (ϵ, σ)
  - Graph is sparse

Outline

Evaluating the Clusters

- How to evaluate the clusters we found?

Evaluating the Clusters

- How to evaluate the clusters we found?
- If labels are not available during evaluation:
  - $\frac{\text{intercluster separation}}{\text{intracluster cohesion}} = \frac{\sum_{i,j}(\boldsymbol{m}_i - \boldsymbol{m}_j)^2}{\sum_i \frac{1}{|G_i|}\sum_{\boldsymbol{x}\in G_i}(\boldsymbol{x}-\boldsymbol{m}_i)^2}$ (the higher the better)
- If labels (i.e., $\{\boldsymbol{r}^{(t)}\}_{t=1}^N$, $\boldsymbol{r}^{(t)} \in \mathbb{R}^K$) are available during the evaluation:
  - $entropy(G_i) = -\sum_{j=1}^K P_i[r_j^{(t)} = 1] \lg P_i[r_j^{(t)} = 1]$, where $P_i[r_j^{(t)} = 1]$ denotes the portions of instances in G_i which belong to class j
    - Here we define $\lg 0 = 0$
  - $entropy_{total}(\mathcal{X}) = \sum_{i=1}^K \frac{|G_i|}{N} entropy(G_i)$ (the lower the better)
- Indirect evaluation: if clustering is used to help perform another task, then we can measure the performance of that task instead
  - E.g., click-through rate of the recommended item in a website (where clustering is used to group similar items/users)

Deciding the Number of Clusters K

- In the previous semiparametric methods, K is determined in advance
  - We can decide K using the cross validation technique
  - Plot the reconstruction error against K and pick the "elbow"
- In hierarchical clustering, K is decided along with h
  - h should be set to cut the "big jump"
- K can be either a parameter or a hyperparameter
- There are extensions for semiparametric methods that adapt K during the iteration
  - E.g.?

Deciding the Number of Clusters K

- In the previous semiparametric methods, K is determined in advance
  - We can decide K using the cross validation technique
  - Plot the reconstruction error against K and pick the "elbow"
- In hierarchical clustering, K is decided along with h
  - h should be set to cut the "big jump"
- K can be either a parameter or a hyperparameter
- There are extensions for semiparametric methods that adapt K during the iteration
  - E.g.? at each iteration, we can drop groups that are too small and/or split groups that are too large

Outline

Why Iterative Methods Work?

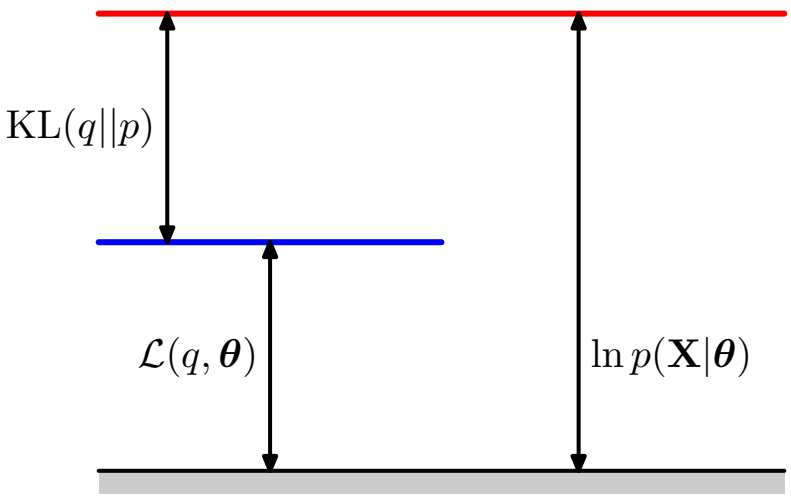
- We have seen the iterative methods for clustering
  - K-means
  - Semiparametric density estimation
- But we haven't answered the following questions:
  - Why does the iteration end?
  - Why is the clusters found in Step 2 better than the ones found in the previous iteration?

Latent Variables and Complete Likelihood (1/2)

- Problem definition: given a dataset $\mathcal{X} = \{\boldsymbol{x}^{(t)}\}_{t=1}^{N}$, suppose $p(\boldsymbol{x}) = \sum_{i=1}^{K} p(\boldsymbol{x}|G_i)P[G_i]$ and denote $\Theta = (\theta_i, \pi_i)_{i=1}^{K}$ where θ_i parametrizes $p(\boldsymbol{x}|G_i)$ and $\pi_i = P[G_i]$, we want to find Θ such that the log likelihood $\ln P[\mathcal{X}|\Theta]$ is maximized
  - $\ln P[\mathcal{X}|\Theta] = \sum_{t=1}^{N} \ln \sum_{i=1}^{K} p(\boldsymbol{x}^{(t)}|\theta_i)\pi_i d\boldsymbol{x}$
  - Unfortunately, since we don't know which instance belongs to which group, we cannot solve this this objective analytically
- Now suppose there is a set $\mathcal{Z} = \{\boldsymbol{z}^{(t)}\}_{t=1}^{N}$ of ***latent variables***, the ***complete likelihood*** can be written as: $\ln P[\mathcal{X}, \mathcal{Z}|\Theta]$
  - $z_i^{(t)} = 1$ if $\boldsymbol{x}^{(t)}$ belongs to group i; 0 otherwise
  - If we have $\mathcal{Z}$, we can solve this objective as we did in the parametric classification
  - Unfortunately, we don't know $\mathcal{Z}$
  - So let's create it and maximize $E_{\mathcal{Z}}[\ln P[\mathcal{X}|\Theta]]$

Latent Variables and Complete Likelihood (2/2)

- Observe that
  $\ln P[\mathcal{X}, \mathcal{Z}|\Theta] = \ln P[\mathcal{Z}|\mathcal{X}, \Theta] + \ln P[\mathcal{X}|\Theta]$
- We have
  $E_{\mathcal{Z}}[\ln P[\mathcal{X}|\Theta]] = L(q, \Theta) + KL(q||P)$ ***for any*** distribution q of $\mathcal{Z}$
  - $L(q, \Theta) = \sum_{\mathcal{Z}} q(\mathcal{Z}) \ln \left(\frac{P[\mathcal{X}, \mathcal{Z}|\Theta]}{q(\mathcal{Z})} \right)$
  - $KL(q||P) = -\sum_{\mathcal{Z}} q(\mathcal{Z}) \ln \left(\frac{P[\mathcal{Z}|\mathcal{X}, \Theta]}{q(\mathcal{Z})} \right)$
- Both $L(q, \Theta)$ and $KL(q||P)$ are ***functional*** of q
- Since $KL(q||P)$ is the ***relative entropy*** (or ***Kullback-Leibler divergence***) and is always greater than 0 [Proof: by Jensen's inequality or $\ln x \leqslant x - 1$], we have the figure at right:

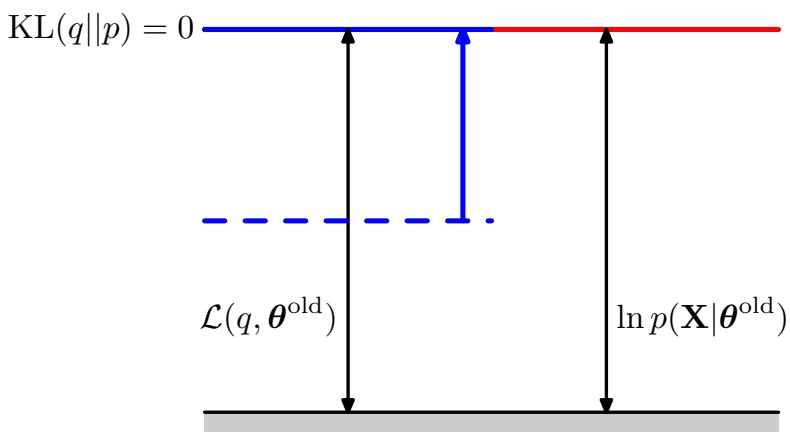


Iterative Methods: A Functional Perspective (1/3)

- To maximize $E_{\mathcal{Z}}[\ln P[\mathcal{X}|\Theta]]$, we can employ an iterative method based on $E_{\mathcal{Z}}[\ln P[\mathcal{X}|\Theta]] = L(q,\Theta) + KL(q||P)$
  - Since q is unknown, we make up q
  - We don't have to make up $\mathcal{Z}$ this time because we try out all possible $\mathcal{Z}$ in $L(q,\Theta)$ and $KL(q||P)$
- Start from a random guess about Θ, iterate the following steps:
  1. Update q based on current Θ such that the blue line is up-aligned with the red
  2. Update Θ based on current q to raises the red line
- Stop until Θ converges
- Why another version?
  - We are sure that $E_{\mathcal{Z}}[\ln P[\mathcal{X}|\Theta]]$ (i.e., read line) can be raised at each iteration (although up to a local optimal)

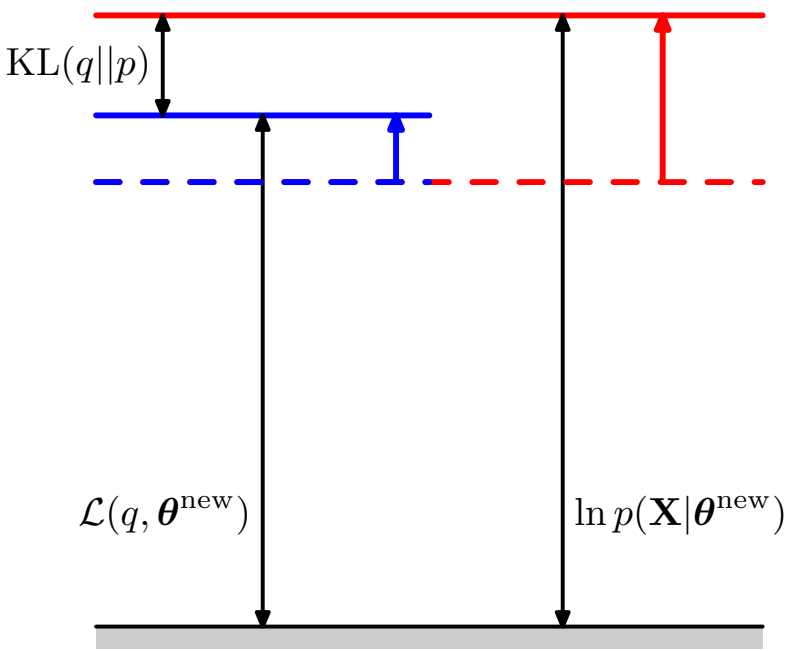
Iterative Methods: A Functional Perspective (2/3)

- Denote by Θ^{old} the parameters found in the previous iteration
- In Step 1, if we update q such that $q(\mathcal{Z}) = P[\mathcal{Z}|\mathcal{X}, \Theta^{old}]$
  - $KL(q||P) = -\sum_{\mathcal{Z}} q(\mathcal{Z}) \ln 1 = 0$
  - $\ln P[\mathcal{X}|\Theta^{old}] = L(q, \Theta^{old}) + 0$
- Note the value of $E_{\mathcal{Z}}[\ln P[\mathcal{X}|\Theta^{old}]]$ won't change as we vary q
- So this step basically raises $L(q, \Theta^{old})$ such that the blue line is up-aligned with the red



Iterative Methods: A Functional Perspective (3/3)

- Fixing q, for any Θ we have
  $E_{\mathcal{Z}}[\ln P[\mathcal{X}|\Theta]] = L(q,\Theta) + KL(q||P) =$
  $\sum_{\mathcal{Z}} P[\mathcal{Z}|\mathcal{X},\Theta^{old}] \ln\left(\frac{P[\mathcal{X},\mathcal{Z}|\Theta]}{P[\mathcal{Z}|\mathcal{X},\Theta^{old}]}\right) -$
  $\sum_{\mathcal{Z}} P[\mathcal{Z}|\mathcal{X},\Theta^{old}] \ln\left(\frac{P[\mathcal{Z}|\mathcal{X},\Theta]}{P[\mathcal{Z}|\mathcal{X},\Theta^{old}]}\right)$
- In Step 2, we find Θ^{new} maximizing $L(q,\Theta)$ (blue line)
  - $KL(q||P) =$
    $-\sum_{\mathcal{Z}} P[\mathcal{Z}|\mathcal{X},\Theta^{old}] \ln\left(\frac{P[\mathcal{Z}|\mathcal{X},\Theta^{new}]}{P[\mathcal{Z}|\mathcal{X},\Theta^{old}]}\right) \geqslant 0$
  - $E_{\mathcal{Z}}[\ln P[\mathcal{X}|\Theta^{new}]] \geqslant E_{\mathcal{Z}}[\ln P[\mathcal{X}|\Theta^{old}]]$
- So this step basically raises the red line, meanwhile leaving the blue behind
- Repeating Steps 1 and 2 lifts $E_{\mathcal{Z}}[\ln P[\mathcal{X}|\Theta]]$ till some local optimum



Outline

Expectation Maximization (1/2)

- Note that in step 1, we don't need to write down q explicitly
  - We just need to evaluate terms in $E_{\mathcal{Z}}[\ln P[\mathcal{X}|\Theta]]$ (a function of Θ to be maximized in step 2) that are related to q
- Fixing $q(\mathcal{Z}) = P[\mathcal{Z}|\mathcal{X}, \Theta^{old}]$, we have

$$L(q, \Theta) = \sum_{\mathcal{Z}} P[\mathcal{Z}|\mathcal{X}, \Theta^{old}] \ln \left(\frac{P[\mathcal{X}, \mathcal{Z}|\Theta]}{P[\mathcal{Z}|\mathcal{X}, \Theta^{old}]} \right)$$
$$= \sum_{\mathcal{Z}} P[\mathcal{Z}|\mathcal{X}, \Theta^{old}] \ln \left(P[\mathcal{X}, \mathcal{Z}|\Theta] \right) - \sum_{\mathcal{Z}} P[\mathcal{Z}|\mathcal{X}, \Theta^{old}] \ln \left(P[\mathcal{Z}|\mathcal{X}, \Theta^{old}] \right)$$
$$= E_{\mathcal{Z}}[\ln \left(P[\mathcal{X}, \mathcal{Z}|\Theta] \right) |\mathcal{X}, \Theta^{old}] + constant$$

  - The second term is an entropy of $\mathcal{Z}$ and is independent of Θ
- Steps 1 and 2 can be refined to be:
  - ***Expectation step (E-step)***: Formulate $\mathcal{Q}(\Theta; \Theta^{old}) = E_{\mathcal{Z}}[\ln \left(P[\mathcal{X}, \mathcal{Z}|\Theta] \right) |\mathcal{X}, \Theta^{old}]$ and evaluate the terms related to $P[\mathcal{Z}|\mathcal{X}, \Theta^{old}]$
  - ***Maximization step (M-step)***: Solve $\Theta^{new} = \arg_{\Theta} \max \mathcal{Q}(\Theta; \Theta^{old})$

Expectation Maximization (2/2)

- The ***Expectation Maximization (EM)*** algorithm is a general technique to find the maximum likelihood solutions for probabilistic models having latent variables
  - Typically, latent variables are discrete, and there is one latent variable per observed instance

Input: $\mathcal{X} \leftarrow \{\boldsymbol{x}^{(t)}\}_{t=1}^{N}$
Output: Θ^{new}, a local optimizer of $E_{\mathcal{Z}}[P[\mathcal{X}|\Theta]]$

Choose an initial Θ^{new};
repeat
  $\Theta^{old} \leftarrow \Theta^{new}$;
  Formulate $\mathcal{Q}(\Theta;\Theta^{old}) = E_{\mathcal{Z}}[\ln(P[\mathcal{X},\mathcal{Z}|\Theta])|\mathcal{X},\Theta^{old}]$ and evaluate the terms related to $P[\mathcal{Z}|\mathcal{X},\Theta^{old}]$; `// E-step`
  $\Theta^{new} \leftarrow \arg_{\Theta}\max \mathcal{Q}(\Theta;\Theta^{old})^{new}$; `// M-step`
until Θ^{new} *converges*;

Algorithm 4: The general EM algorithm.

Outline

I.I.D. Instances

- Assume that $(\boldsymbol{x}^{(t)}, \boldsymbol{z}^{(t)})$ are i.i.d. samples drawn from some distribution
- By definition, $P[\mathcal{X}, \mathcal{Z}] = \Pi_{t=1}^{N} P[\boldsymbol{x}^{(t)}, \boldsymbol{z}^{(t)}]$
- Denote $\boldsymbol{e}_1 = [1, 0 \cdots, 0]^\top, \boldsymbol{e}_2 = [0, 1 \cdots, 0]^\top, \cdots, \boldsymbol{e}_K = [0, 0 \cdots, 1]^\top \in \mathbb{R}^K$
- We have $P[\mathcal{X}] = \Sigma_{\mathcal{Z}} P[\mathcal{X}, \mathcal{Z}] = \Sigma_{\mathcal{Z}} \Pi_{t=1}^{N} P[\boldsymbol{x}^{(t)}, \boldsymbol{z}^{(t)}] = \Sigma_{\boldsymbol{z}^{(1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \cdots \Sigma_{\boldsymbol{z}^{(N)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \Pi_{t=1}^{N} P[\boldsymbol{x}^{(t)}, \boldsymbol{z}^{(t)}] = \Pi_{t=1}^{N} \Sigma_{\boldsymbol{z}^{(t)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} P[\boldsymbol{x}^{(t)}, \boldsymbol{z}^{(t)}] = \Pi_{t=1}^{N} P[\boldsymbol{x}^{(t)}]$
- So, $P[\mathcal{Z}|\mathcal{X}, \Theta^{old}] = \frac{P[\mathcal{X}, \mathcal{Z}|\Theta^{old}]}{P[\mathcal{X}|\Theta^{old}]} = \frac{\Pi_{t=1}^{N} P[\boldsymbol{x}^{(t)}, \boldsymbol{z}^{(t)}|\Theta^{old}]}{\Pi_{t=1}^{N} P[\boldsymbol{x}^{(t)}|\Theta^{old}]} = \Pi_{t=1}^{N} P[\boldsymbol{z}^{(t)}|\boldsymbol{x}^{(t)}, \Theta^{old}]$

Formulating $\mathcal{Q}(\Theta;\Theta^{old})$ (1/3)

- Next, we formulate $\mathcal{Q}(\Theta;\Theta^{old})$ given $P[\mathcal{Z}|\mathcal{X},\Theta^{old}] = \Pi_{t=1}^{N} P[\boldsymbol{z}^{(t)}|\boldsymbol{x}^{(t)},\Theta^{old}]$ (due to the i.i.d. instances), and the assumption of mixture density:
  - $\Theta = (\theta_i,\pi_i)_{i=1}^{K}$ where θ_i parametrizes $p(\boldsymbol{x}|G_i)$ and $\pi_i = P[G_i]$
- Denote by $d(\boldsymbol{z}^{(t)})$ the index of attribute of $\boldsymbol{z}^{(t)}$ equal to 1
  - $P[\boldsymbol{x}^{(t)},\boldsymbol{z}^{(t)}|\Theta] = P[\boldsymbol{x}^{(t)}|\boldsymbol{z}^{(t)},\Theta]P[\boldsymbol{z}^{(t)}|\Theta] = P[\boldsymbol{x}^{(t)}|\boldsymbol{z}^{(t)},\theta_{d(\boldsymbol{z}^{(t)})}]\pi_{d(\boldsymbol{z}^{(t)})}$
- For brevity, we use the shorthand $P[z_i^{(t)}]$ for $P[\boldsymbol{z}^{(t)} = \boldsymbol{e}_i]$ (or equivalently $P[z_i^{(t)} = 1]$)

Formulating $\mathcal{Q}(\Theta;\Theta^{old})$ (2/3)

$$\mathcal{Q}(\Theta;\Theta^{old}) = E_{\mathcal{Z}}[\ln(P[\mathcal{X},\mathcal{Z}|\Theta])|\mathcal{X},\Theta^{old}] = \sum_{\mathcal{Z}} \ln(P[\mathcal{X},\mathcal{Z}|\Theta]) P[\mathcal{Z}|\mathcal{X},\Theta^{old}]$$

Formulating $\mathcal{Q}(\Theta;\Theta^{old})$ (2/3)

$$\mathcal{Q}(\Theta;\Theta^{old}) = E_{\mathcal{Z}}[\ln(P[\mathcal{X},\mathcal{Z}|\Theta])|\mathcal{X},\Theta^{old}] = \sum_{\mathcal{Z}} \ln(P[\mathcal{X},\mathcal{Z}|\Theta]) P[\mathcal{Z}|\mathcal{X},\Theta^{old}]$$
$$= \sum_{\mathcal{Z}} \sum_{t=1}^{N} \ln\left(P[\boldsymbol{x}^{(t)},\boldsymbol{z}^{(t)}|\Theta]\right) \Pi_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$

Formulating $\mathcal{Q}(\Theta;\Theta^{old})$ (2/3)

$$\mathcal{Q}(\Theta;\Theta^{old}) = E_{\mathcal{Z}}[\ln(P[\mathcal{X},\mathcal{Z}|\Theta])|\mathcal{X},\Theta^{old}] = \sum_{\mathcal{Z}} \ln(P[\mathcal{X},\mathcal{Z}|\Theta]) P[\mathcal{Z}|\mathcal{X},\Theta^{old}]$$
$$= \sum_{\mathcal{Z}} \sum_{t=1}^{N} \ln\left(P[\boldsymbol{x}^{(t)},\boldsymbol{z}^{(t)}|\Theta]\right) \Pi_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$
$$= \sum_{\mathcal{Z}} \sum_{t=1}^{N} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{z}^{(t)},\theta_{d(\boldsymbol{z}^{(t)})}]\pi_{d(\boldsymbol{z}^{(t)})}\right) \Pi_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$

Formulating $\mathcal{Q}(\Theta;\Theta^{old})$ (2/3)

$$\mathcal{Q}(\Theta;\Theta^{old}) = E_{\mathcal{Z}}[\ln(P[\mathcal{X},\mathcal{Z}|\Theta])|\mathcal{X},\Theta^{old}] = \sum_{\mathcal{Z}} \ln(P[\mathcal{X},\mathcal{Z}|\Theta]) P[\mathcal{Z}|\mathcal{X},\Theta^{old}]$$
$$= \sum_{\mathcal{Z}} \sum_{t=1}^{N} \ln\left(P[\boldsymbol{x}^{(t)},\boldsymbol{z}^{(t)}|\Theta]\right) \Pi_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$
$$= \sum_{\mathcal{Z}} \sum_{t=1}^{N} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{z}^{(t)},\theta_{d(\boldsymbol{z}^{(t)})}]\pi_{d(\boldsymbol{z}^{(t)})}\right) \Pi_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$
$$= \sum_{\mathcal{Z}} \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \delta_{\boldsymbol{z}^{(t)},\boldsymbol{e}} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right)$$
$$\Pi_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}] \quad // \ \delta_{a,b} = 1 \text{if } a = b; \ 0 \text{otherwise}$$

Formulating $\mathcal{Q}(\Theta;\Theta^{old})$ (2/3)

$$\mathcal{Q}(\Theta;\Theta^{old}) = E_{\mathcal{Z}}[\ln(P[\mathcal{X},\mathcal{Z}|\Theta])|\mathcal{X},\Theta^{old}] = \sum_{\mathcal{Z}} \ln(P[\mathcal{X},\mathcal{Z}|\Theta]) P[\mathcal{Z}|\mathcal{X},\Theta^{old}]$$
$$= \sum_{\mathcal{Z}} \sum_{t=1}^{N} \ln\left(P[\boldsymbol{x}^{(t)},\boldsymbol{z}^{(t)}|\Theta]\right) \Pi_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$
$$= \sum_{\mathcal{Z}} \sum_{t=1}^{N} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{z}^{(t)},\theta_{d(\boldsymbol{z}^{(t)})}]\pi_{d(\boldsymbol{z}^{(t)})}\right) \Pi_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$
$$= \sum_{\mathcal{Z}} \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \delta_{\boldsymbol{z}^{(t)},\boldsymbol{e}} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right)$$
$$\Pi_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}] \quad // \ \delta_{a,b} = 1 \text{if } a = b; \ 0 \text{otherwise}$$
$$= \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right) \sum_{\mathcal{Z}} \delta_{\boldsymbol{z}^{(t)},\boldsymbol{e}} \Pi_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$

Formulating $\mathcal{Q}(\Theta;\Theta^{old})$ (2/3)

$$\mathcal{Q}(\Theta;\Theta^{old}) = E_{\mathcal{Z}}[\ln(P[\mathcal{X},\mathcal{Z}|\Theta])|\mathcal{X},\Theta^{old}] = \sum_{\mathcal{Z}} \ln(P[\mathcal{X},\mathcal{Z}|\Theta]) P[\mathcal{Z}|\mathcal{X},\Theta^{old}]$$
$$= \sum_{\mathcal{Z}} \sum_{t=1}^{N} \ln\left(P[\boldsymbol{x}^{(t)},\boldsymbol{z}^{(t)}|\Theta]\right) \prod_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$
$$= \sum_{\mathcal{Z}} \sum_{t=1}^{N} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{z}^{(t)},\theta_{d(\boldsymbol{z}^{(t)})}]\pi_{d(\boldsymbol{z}^{(t)})}\right) \prod_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$
$$= \sum_{\mathcal{Z}} \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \delta_{\boldsymbol{z}^{(t)},\boldsymbol{e}} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right)$$
$$\prod_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}] \quad // \ \delta_{a,b} = 1 \text{if } a = b;\ 0 \text{otherwise}$$
$$= \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right) \sum_{\mathcal{Z}} \delta_{\boldsymbol{z}^{(t)},\boldsymbol{e}} \prod_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$
$$= \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right)$$
$$\sum_{\boldsymbol{z}^{(1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \cdots \sum_{\boldsymbol{z}^{(N)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \delta_{\boldsymbol{z}^{(t)},\boldsymbol{e}} \prod_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$

Formulating $\mathcal{Q}(\Theta;\Theta^{old})$ (2/3)

$$\mathcal{Q}(\Theta;\Theta^{old}) = E_{\mathcal{Z}}[\ln(P[\mathcal{X},\mathcal{Z}|\Theta])|\mathcal{X},\Theta^{old}] = \sum_{\mathcal{Z}} \ln(P[\mathcal{X},\mathcal{Z}|\Theta]) P[\mathcal{Z}|\mathcal{X},\Theta^{old}]$$
$$= \sum_{\mathcal{Z}} \sum_{t=1}^{N} \ln\left(P[\boldsymbol{x}^{(t)},\boldsymbol{z}^{(t)}|\Theta]\right) \prod_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$
$$= \sum_{\mathcal{Z}} \sum_{t=1}^{N} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{z}^{(t)},\theta_{d(\boldsymbol{z}^{(t)})}]\pi_{d(\boldsymbol{z}^{(t)})}\right) \prod_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$
$$= \sum_{\mathcal{Z}} \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \delta_{\boldsymbol{z}^{(t)},\boldsymbol{e}} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right)$$
$$\prod_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}] \quad // \ \delta_{a,b} = 1 \text{if } a = b;\ 0 \text{otherwise}$$
$$= \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right) \sum_{\mathcal{Z}} \delta_{\boldsymbol{z}^{(t)},\boldsymbol{e}} \prod_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$
$$= \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right)$$
$$\sum_{\boldsymbol{z}^{(1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \cdots \sum_{\boldsymbol{z}^{(N)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \delta_{\boldsymbol{z}^{(t)},\boldsymbol{e}} \prod_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$
$$= \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right) \sum_{\boldsymbol{z}^{(1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \cdots \sum_{\boldsymbol{z}^{(t-1)}=\boldsymbol{e}_1}$$
$$\sum_{\boldsymbol{z}^{(t+1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \cdots \sum_{\boldsymbol{z}^{(N)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \prod_{j=1,j\neq t}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}] P[\boldsymbol{e}|\boldsymbol{x}^{(t)},\Theta^{old}]$$

Formulating $\mathcal{Q}(\Theta;\Theta^{old})$ (2/3)

$$\mathcal{Q}(\Theta;\Theta^{old}) = E_{\mathcal{Z}}[\ln(P[\mathcal{X},\mathcal{Z}|\Theta])|\mathcal{X},\Theta^{old}] = \sum_{\mathcal{Z}} \ln(P[\mathcal{X},\mathcal{Z}|\Theta]) P[\mathcal{Z}|\mathcal{X},\Theta^{old}]$$
$$= \sum_{\mathcal{Z}} \sum_{t=1}^{N} \ln\left(P[\boldsymbol{x}^{(t)},\boldsymbol{z}^{(t)}|\Theta]\right) \prod_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$
$$= \sum_{\mathcal{Z}} \sum_{t=1}^{N} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{z}^{(t)},\theta_{d(\boldsymbol{z}^{(t)})}]\pi_{d(\boldsymbol{z}^{(t)})}\right) \prod_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$
$$= \sum_{\mathcal{Z}} \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \delta_{\boldsymbol{z}^{(t)},\boldsymbol{e}} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right)$$
$$\prod_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}] \quad // \ \delta_{a,b} = 1 \text{if } a = b; \ 0 \text{otherwise}$$
$$= \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right) \sum_{\mathcal{Z}} \delta_{\boldsymbol{z}^{(t)},\boldsymbol{e}} \prod_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$
$$= \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right)$$
$$\sum_{\boldsymbol{z}^{(1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \cdots \sum_{\boldsymbol{z}^{(N)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \delta_{\boldsymbol{z}^{(t)},\boldsymbol{e}} \prod_{j=1}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]$$
$$= \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right) \sum_{\boldsymbol{z}^{(1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \cdots \sum_{\boldsymbol{z}^{(t-1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K}$$
$$\sum_{\boldsymbol{z}^{(t+1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \cdots \sum_{\boldsymbol{z}^{(N)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \prod_{j=1,j\neq t}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}] P[\boldsymbol{e}|\boldsymbol{x}^{(t)},\Theta^{old}]$$
$$= \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right) \left(\sum_{\boldsymbol{z}^{(1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \cdots \sum_{\boldsymbol{z}^{(t-1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K}\right.$$
$$\left.\sum_{\boldsymbol{z}^{(t+1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \cdots \sum_{\boldsymbol{z}^{(N)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \prod_{j=1,j\neq t}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]\right) P[\boldsymbol{e}|\boldsymbol{x}^{(t)},\Theta^{old}]$$

Formulating $\mathcal{Q}(\Theta;\Theta^{old})$ (3/3)

$$= \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right)\left(\sum_{\boldsymbol{z}^{(1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K}\cdots\sum_{\boldsymbol{z}^{(t-1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K}\right.$$
$$\left.\sum_{\boldsymbol{z}^{(t+1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K}\cdots\sum_{\boldsymbol{z}^{(N)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K}\prod_{j=1,j\neq t}^{N}P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]\right)P[\boldsymbol{e}|\boldsymbol{x}^{(t)},\Theta^{old}]$$

Formulating $\mathcal{Q}(\Theta;\Theta^{old})$ (3/3)

$$= \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right)\left(\sum_{\boldsymbol{z}^{(1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K}\cdots\sum_{\boldsymbol{z}^{(t-1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K}\right.$$
$$\left.\sum_{\boldsymbol{z}^{(t+1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K}\cdots\sum_{\boldsymbol{z}^{(N)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K}\prod_{j=1,j\neq t}^{N}P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]\right)P[\boldsymbol{e}|\boldsymbol{x}^{(t)},\Theta^{old}]$$
$$= \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right)$$
$$\left(\prod_{j=1,j\neq t}^{N}\sum_{\boldsymbol{z}^{(j)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K}P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]\right)P[\boldsymbol{e}|\boldsymbol{x}^{(t)},\Theta^{old}]$$

Formulating $\mathcal{Q}(\Theta;\Theta^{old})$ (3/3)

$$= \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right) \left(\sum_{\boldsymbol{z}^{(1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \cdots \sum_{\boldsymbol{z}^{(t-1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \sum_{\boldsymbol{z}^{(t+1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \cdots \sum_{\boldsymbol{z}^{(N)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \prod_{j=1,j\neq t}^{N} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]\right) P[\boldsymbol{e}|\boldsymbol{x}^{(t)},\Theta^{old}]$$

$$= \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right) \left(\prod_{j=1,j\neq t}^{N} \sum_{\boldsymbol{z}^{(j)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]\right) P[\boldsymbol{e}|\boldsymbol{x}^{(t)},\Theta^{old}]$$

$$= \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right) P[\boldsymbol{e}|\boldsymbol{x}^{(t)},\Theta^{old}]$$

Formulating $\mathcal{Q}(\Theta;\Theta^{old})$ (3/3)

$$
\begin{aligned}
&= \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right)\left(\sum_{\boldsymbol{z}^{(1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K}\cdots\sum_{\boldsymbol{z}^{(t-1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K}\right.\\
&\qquad\left.\sum_{\boldsymbol{z}^{(t+1)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K}\cdots\sum_{\boldsymbol{z}^{(N)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K}\prod_{j=1,j\neq t}^{N}P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]\right)P[\boldsymbol{e}|\boldsymbol{x}^{(t)},\Theta^{old}]\\
&= \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right)\\
&\qquad\left(\prod_{j=1,j\neq t}^{N}\sum_{\boldsymbol{z}^{(j)}=\boldsymbol{e}_1}^{\boldsymbol{e}_K}P[\boldsymbol{z}^{(j)}|\boldsymbol{x}^{(j)},\Theta^{old}]\right)P[\boldsymbol{e}|\boldsymbol{x}^{(t)},\Theta^{old}]\\
&= \sum_{t=1}^{N} \sum_{\boldsymbol{e}=\boldsymbol{e}_1}^{\boldsymbol{e}_K} \ln\left(P[\boldsymbol{x}^{(t)}|\boldsymbol{e},\theta_{d(\boldsymbol{e})}]\pi_{d(\boldsymbol{e})}\right)P[\boldsymbol{e}|\boldsymbol{x}^{(t)},\Theta^{old}]\\
&= \sum_{t=1}^{N} \sum_{i=1}^{K} \ln\left(\pi_i\right)P[z_i^{(t)}|\boldsymbol{x}^{(t)},\Theta^{old}]+\\
&\qquad\sum_{t=1}^{N} \sum_{i=1}^{K} \ln\left(P[\boldsymbol{x}^{(t)}|z_i^{(t)},\theta_i]\right)P[z_i^{(t)}|\boldsymbol{x}^{(t)},\Theta^{old}]
\end{aligned}
$$

Evaluating $P[\mathcal{Z}|\mathcal{X}, \Theta^{old}]$

- Given mixtures of i.i.d. samples, we have
  $\mathcal{Q}(\Theta; \Theta^{old}) = \sum_{t=1}^{N} \sum_{i=1}^{K} \ln(\pi_i) P[z_i^{(t)}|\boldsymbol{x}^{(t)}, \Theta^{old}] +$
  $\sum_{t=1}^{N} \sum_{i=1}^{K} \ln\left(P[\boldsymbol{x}^{(t)}|z_i^{(t)}, \theta_i]\right) P[z_i^{(t)}|\boldsymbol{x}^{(t)}, \Theta^{old}]$
- The problem evaluating $P[\mathcal{Z}|\mathcal{X}, \Theta^{old}]$ is thus reduced to evaluating $P[z_i^{(t)}|\boldsymbol{x}^{(t)}, \Theta^{old}]$ for all $1 \leqslant i \leqslant K$ and $1 \leqslant t \leqslant N$

Outline

Evaluating $P[z_i^{(t)}|\boldsymbol{x}^{(t)}, \Theta^{old}]$

- Problem: given Θ^{old}, evaluate $P[z_i^{(t)}|\boldsymbol{x}^{(t)}, \Theta^{old}]$ for all $1 \leqslant i \leqslant K$ and $1 \leqslant t \leqslant N$
- From Bayes' theorem,

$$P[z_i^{(t)}|\boldsymbol{x}^{(t)}, \Theta^{old}] = \frac{P[\boldsymbol{x}^{(t)}|z_i^{(t)}, \Theta^{old}]P[z_i^{(t)}|\Theta^{old}]}{\sum_{j=1}^{K} P[\boldsymbol{x}^{(t)}|z_j^{(t)}, \Theta^{old}]P[z_j^{(t)}|\Theta^{old}]} = \frac{P[\boldsymbol{x}^{(t)}|z_i^{(t)}, \theta_i^{old}]\pi_i^{old}}{\sum_{j=1}^{K} P[\boldsymbol{x}^{(t)}|z_j^{(t)}, \theta_j^{old}]\pi_j^{old}}$$

- If we further assume that instances in each group are normally distributed, then $\theta_i^{old} = (\boldsymbol{\mu}_i^{old}, \boldsymbol{\Sigma}_i^{old})$ and we can easily obtain $P[\boldsymbol{x}^{(t)}|z_i^{(t)}, \theta_i^{old}]$ based on the normal distribution
- For brevity, we denote the evaluated $P[z_i^{(t)}|\boldsymbol{x}^{(t)}, \Theta^{old}]$ by $h_i^{(t)}$
  - $h_i^{(t)}$ aligns with the soft label $z_i^{(t)}$ in semiparametric density estimation

Solving $\arg_{\Theta} \max \mathcal{Q}(\Theta; \Theta^{old})$

- $\mathcal{Q}(\Theta; \Theta^{old}) = \sum_{t=1}^{N} \sum_{i=1}^{K} \ln(\pi_i) h_i^{(t)} + \sum_{t=1}^{N} \sum_{i=1}^{K} \ln\left(P[\boldsymbol{x}^{(t)}|z_i^{(t)}, \theta_i]\right) h_i^{(t)}$
- Observe that the first term of $\mathcal{Q}(\Theta; \Theta^{old})$ depends only on $\{\pi_i\}_{i=1}^{K}$; and the second depends only on $\{\theta_i\}_{i=1}^{K}$
- We can obtain Θ^{new} by solving the two problems individually:
  - $\arg_{\pi_1,\cdots,\pi_K} \max \sum_{t=1}^{N} \sum_{i=1}^{K} \ln(\pi_i) h_i^{(t)}$ subject to $\sum_{i=1}^{K} \pi_i = 1$
  - $\arg_{\theta_1,\cdots,\theta_K} \max \sum_{t=1}^{N} \sum_{i=1}^{K} \ln\left(P[\boldsymbol{x}^{(t)}|z_i^{(t)}, \theta_i]\right) h_i^{(t)}$

Solving $\{\pi\}_{i=1}^{K}$

- Lagrangian:
  $L(\{\boldsymbol{x}^{(t)}\}_{t=1}^{N}, \{\pi_i\}_{i=1}^{K}, \alpha) = \sum_{t=1}^{N}\sum_{i=1}^{K} \ln(\pi_i)\, h_i^{(t)} - \alpha\left(\sum_{i=1}^{K}\pi_i - 1\right)$
- Taking the partial derivatives of L with respect to $\alpha, \pi_1, \cdots, \pi_K$ and setting them to zero we have $\sum_{i=1}^{K}\pi_i = 1$ and
  $\sum_{t=1}^{N}\frac{1}{\pi_i}h_i^{(t)} - \alpha = 0 \Rightarrow \sum_{t=1}^{N} h_i^{(t)} = \pi_i\alpha$ for $i = 1, \cdots, K$
- Summing the equations with α above we have $\sum_{i=1}^{K}\sum_{t=1}^{N} h_i^{(t)} = \sum_{i=1}^{K}\pi_i\alpha \Rightarrow \alpha = \sum_{t=1}^{N}\sum_{i=1}^{K} h_i^{(t)} = \sum_{t=1}^{N}\sum_{i=1}^{K} P[z_i^{(t)}|\boldsymbol{x}^{(t)}, \Theta^{old}] = N$
- Substituting N for α in each of the above equation we have
  $\pi_i = \frac{\sum_{t=1}^{N} h_i^{(t)}}{N}$
  - This aligns with the π_i in semiparametric density estimation

Solving $\{\theta_i\}_{i=1}^K$ (1/2)

- Objective: $\arg_{\theta_1,\cdots,\theta_K} \max \sum_{t=1}^N \sum_{i=1}^K \ln\left(P[\boldsymbol{x}^{(t)}|z_i^{(t)},\theta_i]\right) h_i^{(t)}$
- Since the groups in the mixtures are independent with each other, we can solve $\theta_i = (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ one by one
  - $\arg_{\theta_i} \max \sum_{t=1}^N \ln\left(P[\boldsymbol{x}^{(t)}|z_i^{(t)},\theta_i]\right) h_i^{(t)}$
- With the Gaussian mixture, we have
  $\sum_{t=1}^N \ln\left(P[\boldsymbol{x}^{(t)}|z_i^{(t)},\theta_i]\right) h_i^{(t)} = -\frac{N_i d}{2}\log(2\pi) - \frac{N_i}{2}\log(det(\boldsymbol{\Sigma}_i)) -$
  $\frac{1}{2}\sum_{t=1}^N h_i^{(t)}(\boldsymbol{x}^{(t)}-\boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}^{(t)}-\boldsymbol{\mu}_i) = -\frac{N_i d}{2}\log(2\pi) +$
  $\frac{N_i}{2}\log(det(\boldsymbol{\Sigma}_i^{-1})) - \frac{1}{2}\sum_{t=1}^N h_i^{(t)} tr\left(\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}^{(t)}-\boldsymbol{\mu}_i)(\boldsymbol{x}^{(t)}-\boldsymbol{\mu}_i)^\top\right)$, where
  $N_i = \sum_{t=1}^N h_i^{(t)}$

Solving $\{\theta_i\}_{i=1}^K$ (2/2)

- Taking the partial derivatives of the above objective with respect to $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i^{-1}$ and setting them to zero we have

$$\begin{cases} \sum_{t=1}^{N} h_i^{(t)} (\boldsymbol{x}^{(t)} - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1} = \mathbf{0}^\top \\ \frac{N_i}{2} \boldsymbol{\Sigma}_i - \frac{1}{2} \sum_{t=1}^{N} h_i^{(t)} (\boldsymbol{x}^{(t)} - \boldsymbol{\mu}_i)(\boldsymbol{x}^{(t)} - \boldsymbol{\mu}_i)^\top = \boldsymbol{O} \end{cases}$$

  - $\boldsymbol{m}_i = \frac{\sum_{t=1}^{N} \boldsymbol{x}^{(t)} h_i^{(t)}}{\sum_{t=1}^{N} h_i^{(t)}}$
  - $\boldsymbol{S}_i = \frac{\sum_{t=1}^{N} (\boldsymbol{x}^{(t)} - \boldsymbol{m}_i)(\boldsymbol{x}^{(t)} - \boldsymbol{m}_i)^\top h_i^{(t)}}{\sum_{t=1}^{N} h_i^{(t)}}$

- Again, these results align with the $\boldsymbol{m}_i$ and $\boldsymbol{S}_i$ in semiparametric density estimation

Remarks

- Both the K-means and semiparametric density estimation are EM algorithms
  - The iteration ends and Θ converges to a local optimum
- In particular, when assuming that the priors π_i are all equal and $\boldsymbol{\Sigma}_i = \sigma^2 \boldsymbol{I}$, we have
  - $h_i^{(t)} = \frac{exp\left[-(1/2(s^{old})^2)\left\|\boldsymbol{x}^{(t)} - \boldsymbol{m}_i^{old}\right\|\right]}{\sum_{j=1}^{K} exp\left[-(1/2(s^{old})^2)\left\|\boldsymbol{x}^{(t)} - \boldsymbol{m}_j^{old}\right\|\right]}$
  - The objective $\arg_{\theta_1,\cdots,\theta_K} \max \sum_{t=1}^{N} \sum_{i=1}^{K} \ln\left(P[\boldsymbol{x}^{(t)}|z_i^{(t)}, \theta_i]\right) h_i^{(t)}$ can be rewritten as $\arg_{\boldsymbol{m}_1,\cdots,\boldsymbol{m}_K,s} \min \sum_{t=1}^{N} \sum_{i=1}^{K} \frac{\left\|\boldsymbol{x}^{(t)} - \boldsymbol{m}_i\right\|^2}{s^2} h_i^{(t)}$
  - This is equivalent to minimizing the reconstruction error in the K-means