12. Unsupervised Learning: Clustering

Dongwoo Kim

dongwoo.kim@postech.ac.kr

CSED515 - 2023 Spring

Unsupervised Learning

We've focused on supervised learning tasks, in particular, regression and classification. In next several lectures, we will study about unsupervised learning:

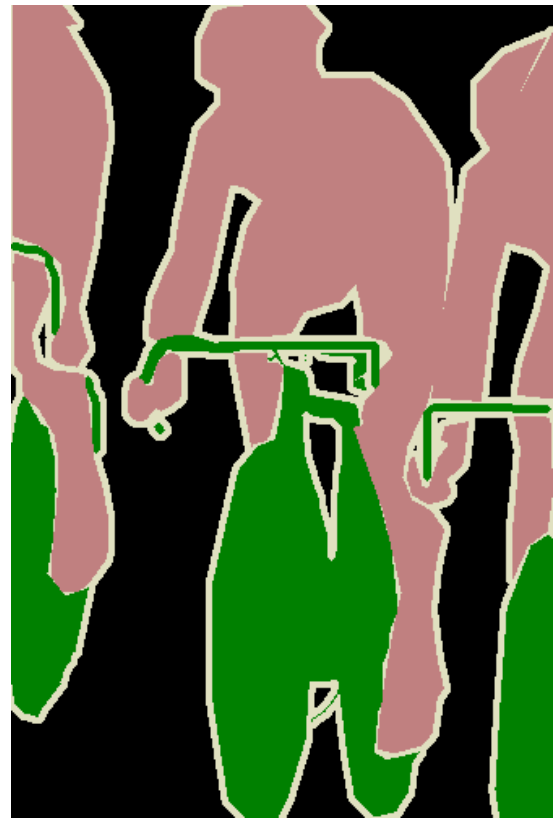
- Clustering (this lecture), e.g., image segmentation
- Feature selection or dimensionality reduction, e.g., PCA
- Generative model, e.g., GAN

Clustering

A famous task of unsupervised learning is the problem of partitioning a set of unlabeled data points into a pre-specified number of groups of similar points using some similarity or distance measure, e.g., image segmentation

Image Segmentation via Classification

In supervised learning setting, we are given human labels regarding segmentation:



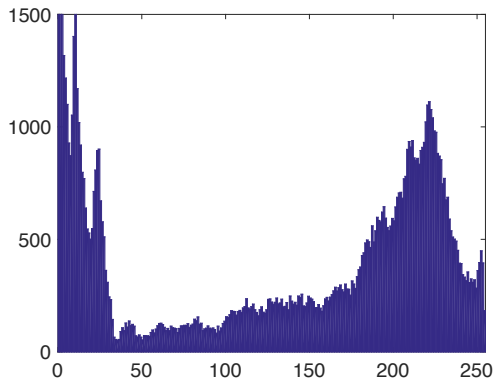
It requires a tremendous amount of human labor!

Image Segmentation via Clustering

Even if we don't have labels, it is possible to perform some segmentation in unsupervised learning as follows:

Image

Intensities



Clustering

Outline

Clustering

1 Algorithms: K-means and Soft K-means

2 Analysis: Gaussian Mixture Model (GMM)

Clustering

A famous task of unsupervised learning is the problem of partitioning a set of unlabeled data points into a pre-specified number of groups of similar points using some similarity or distance measure:

- Nonparametric approach: K-means clustering or soft K-means clustering
- Parametric approach: Expectation-Maximization (EM) algorithm with Gaussian mixture model (GMM) a.k.a. Mixture of Gaussian (MoG)

A Cost Function for Clustering

Given a number $K \in \mathbb{N}$ and a set of data points $\{x_i \in \mathbb{R}^D\}_{i \in [N]}$:

- Find assignment vector $\boldsymbol{r}$:

$$\min_{\boldsymbol{\mu}} \min_{\boldsymbol{r}} \sum_{i \in [N]} \sum_{k \in [K]} r_{ik} \|x_i - \mu_k\|^2 ,$$

$$\text{s.t. } r_{ik} \in \{0, 1\} \ \forall i, k \quad \text{and} \quad \sum_{k \in [K]} r_{ik} = 1 \quad \forall i$$

  where μ_k is a reference point of cluster k

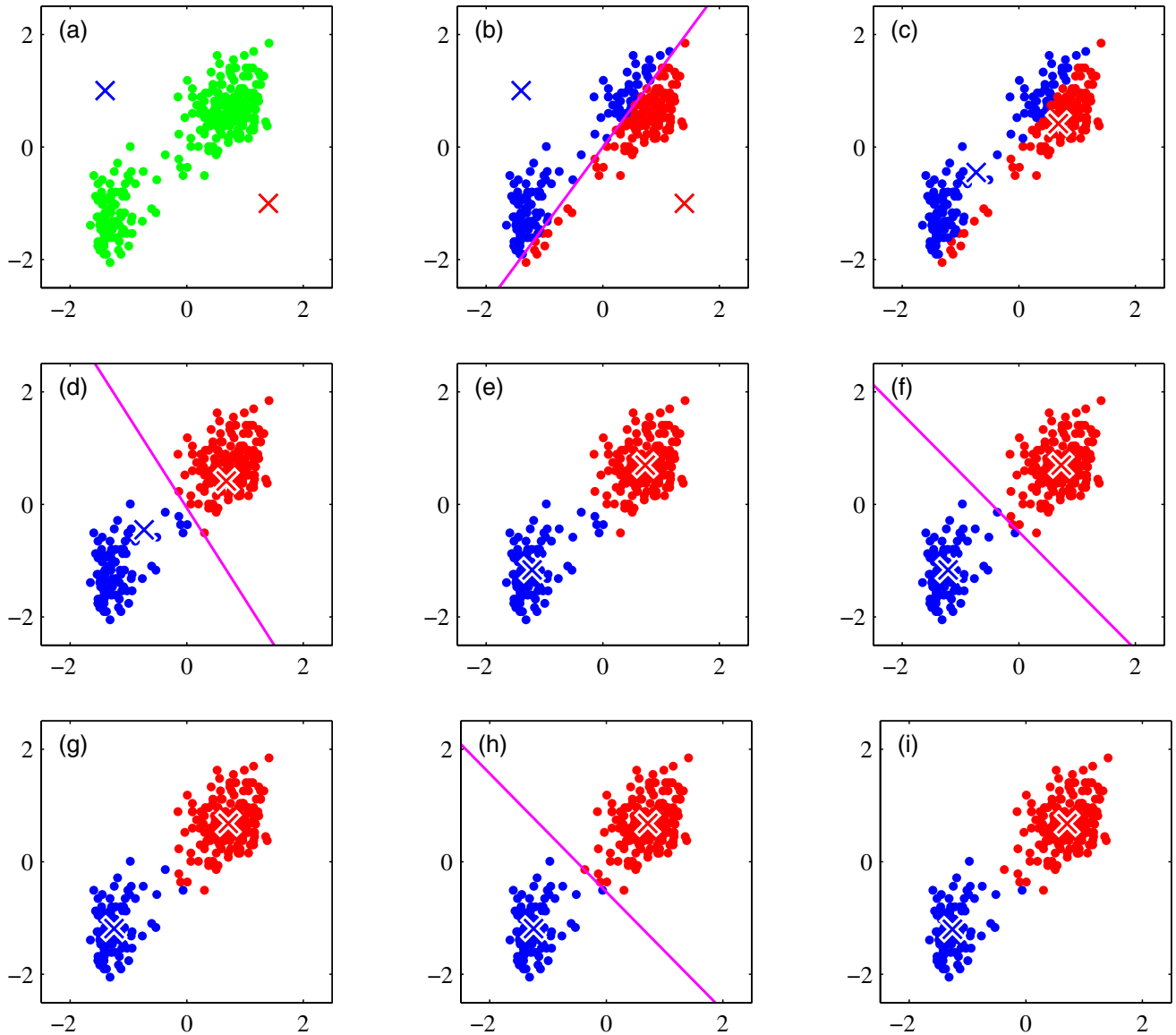
- Similarity measure = Euclidean distance (or some other metric)

K-mean Algorithm (Informal description)

Starting from randomly chosen K centroids $\boldsymbol{\mu} = \{\mu_k\}_{k=1}^K$,

- Assignment step: Given $\boldsymbol{\mu}$, find optimal assignment $\boldsymbol{r}$
- Update step: Given $\boldsymbol{r}$, find centroid $\boldsymbol{\mu}$
- Repeat these two steps until convergence.

Example with 2 Clusters



A data point = RGB information of each pixel

K-means Algorithm

Starting from randomly chosen K centroids $\{\mu_k\}$,

- Assignment step: given $\boldsymbol{\mu}$,

$$r_{ik} = \begin{cases} 1 & \text{if } k = \arg\min_k \|x_i - \mu_k\|_2^2 \\ 0 & \text{otherwise} \end{cases},$$

  i.e., assign a data point to the cluster of the closest reference point

- Update step: given $\boldsymbol{r}$,

$$\mu_k = \frac{\sum_i r_{ik} x_i}{\sum_i r_{ik}} \quad \text{a.k.a. centroid}$$

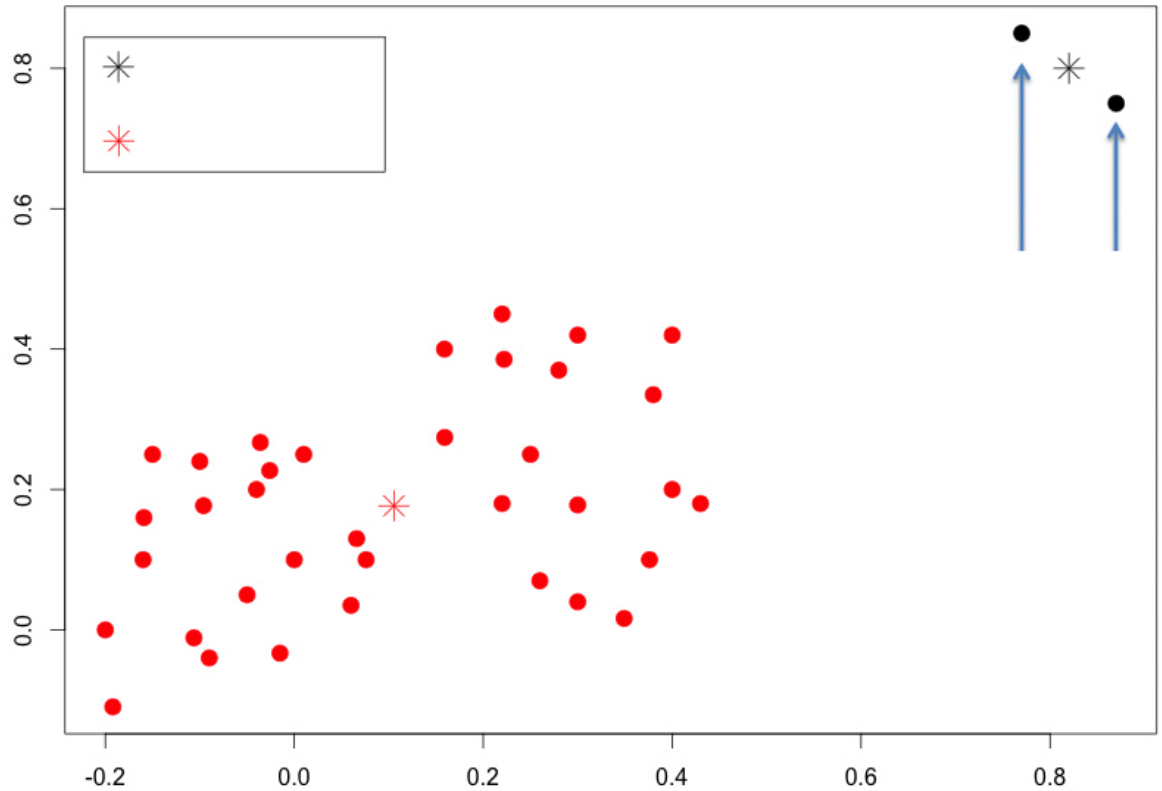
  which is from taking gradient of cost function w.r.t. $\boldsymbol{\mu}$ and setting it to zero

- Repeat these two steps until convergence

Properties of K-means

- Local optimum is found
- Convergence guarantee in a finite number of iteration
- Computational complexity per iteration:
  - Assignment: $O(KND)$
  - Centroid: $O(N)$
- c.f., a set of variants, e.g., K-means++

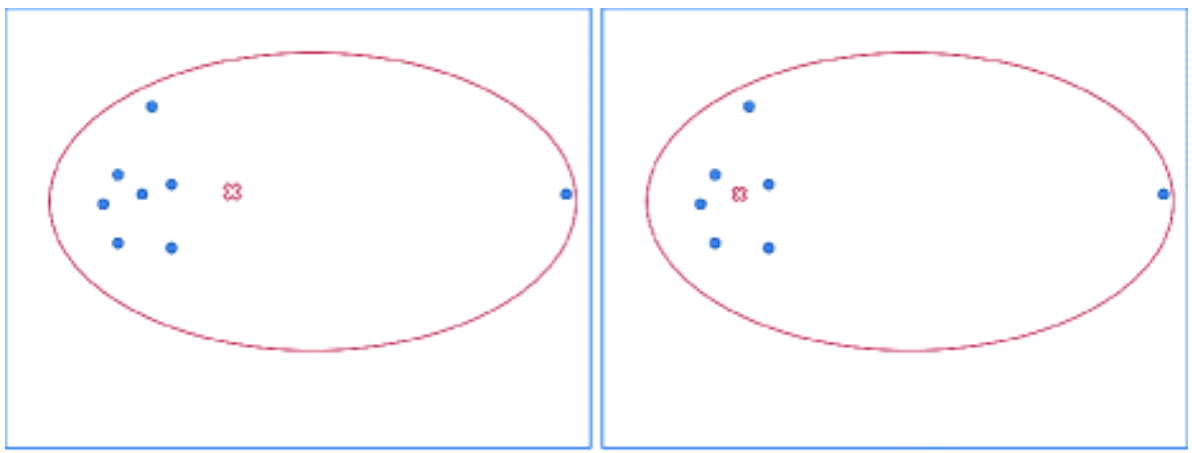
Limitation of K-means



- K-means is sensitive to outliers.

K-medoid

- K-medoids chooses actual data points as centers
- more robust than k-means in presence of outliers
- PAM (Partitioning Around Medoids) is a classic algorithm for k-medoid clustering



(a) Mean (b) Medoid

Soft K-means

- Assignment step: each data point x_i has a soft degree of assignment or responsibility r_{ik} to each cluster k:

$$r_{ik} = \frac{\exp\left(-\beta \|x_i - \mu_k\|^2\right)}{\sum_{\ell \in [K]} \exp\left(-\beta \|x_i - \mu_\ell\|^2\right)} \, .$$

- Update step: Compute new centroids for each cluster, i.e.,

$$\mu_k = \frac{\sum_{i \in [N]} r_{ik} x_i}{\sum_{i \in [N]} r_{ik}} \, .$$

Interpretation of Soft K-means

A corresponding optimization:

$$\text{minimize} \quad \mathcal{J}_{\text{soft}} = \sum_{i \in [N]} \sum_{k \in [K]} r_{ik} \|x_i - \mu_k\|^2 - \frac{1}{\beta} \sum_{i \in [N]} \sum_{k \in [K]} r_{ik} \log r_{ik}$$

- Note that the entropy $H(X) := -\sum_x P(X = x) \log P(X = x)$ quantifies the degree of randomness, e.g., a constant has zero entropy.
- The second term (or entropy) encourages the soft assignments to spread over clusters, and β is the control knob such that smaller β implies softer assignments.

Outline

Clustering

1 Algorithms: K-means and Soft K-means

2 Analysis: Gaussian Mixture Model (GMM)

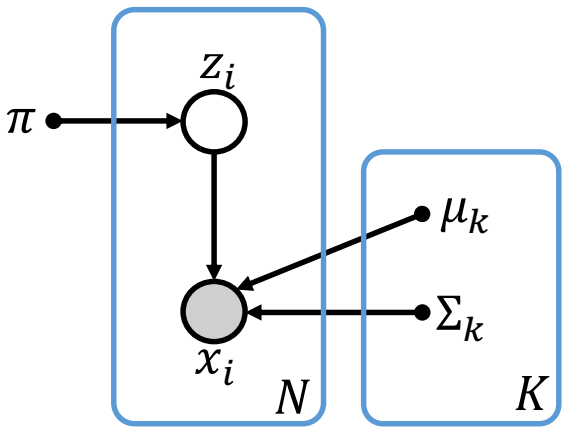
Finite Mixture Model

A semi-parametric model in the form of:

$$p(x) = \sum_{k \in [K]} p(x, z = k) = \sum_{k \in [K]} p_k(x) p(z = k)$$

- A point is drawn from one of K component distributions $\{p_k(\cdot)\}_{k \in [K]}$
- z is the latent variable indicating from which component distribution the point is originated.
- $\{\pi_k := p(z = k)\}_{k \in [K]}$ are the mixing parameters such that $\sum_{k \in [K]} \pi_k = 1$ and $\pi_k \in [0, 1]$.

Gaussian Mixture Model: Graphical Representation



Gaussian Mixture Model (GMM)

- Point x_i's true cluster $z_i \in [K]$ is hidden and independently drawn from

$$p(z_i) = \prod_{k \in [K]} \pi_k^{\mathbb{1}[z_i = k]} \text{, i.e., } p(z_i = k) = \pi_k.$$

- The distribution over observed variables conditioned on the latent variables is

$$p(x_i \mid z_i) = \prod_{k \in [K]} \left(\mathcal{N}(x_i \mid \mu_k, \Sigma_k)\right)^{\mathbb{1}[z_i = k]}$$

or $p(x_i \mid \ell) = \mathcal{N}(x_i \mid \mu_\ell, \Sigma_\ell)$ if $z_i = \ell$.

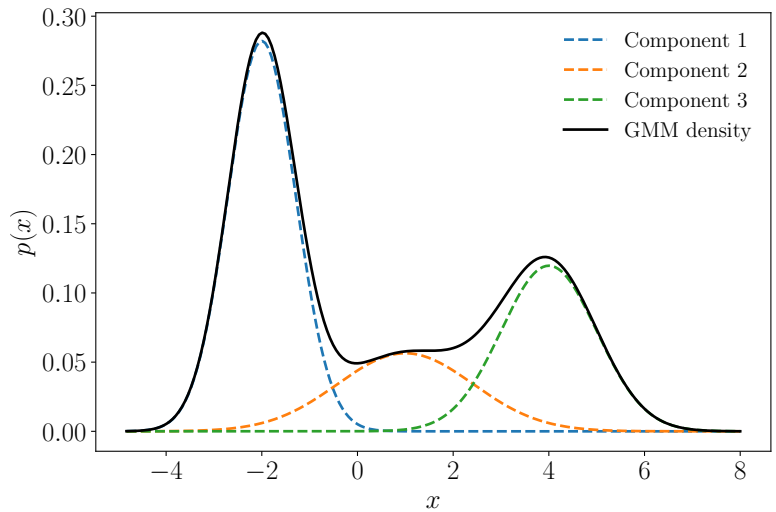
Gaussian Mixture Models: Intuition

- Marginal is a multimodal distribution

$$p(x|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

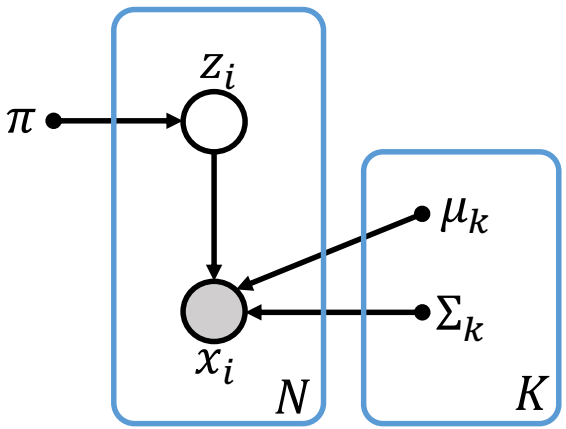
where $\theta = \{\mu_k, \Sigma_k, \pi_k\}_{k=1}^{K}$.

- Flexible than K-means.



$0.5\mathcal{N}(-2, \frac{1}{2}) + 0.2\mathcal{N}(1, 2) + 0.3\mathcal{N}(4, 1)$

Learning GMM



- Compute maximum likelihood estimates of parameters

$$\theta = \{\pi_k, (\mu_k, \Sigma_k)\}_{k \in [K]}$$

- Compute the posterior on latent z_i

$$r_{ik} = p(z_i = k \mid x_i)$$

- Can optimize θ with gradient methods from marginal distribution $p(x|\theta)$?

Parameter estimation via MLE

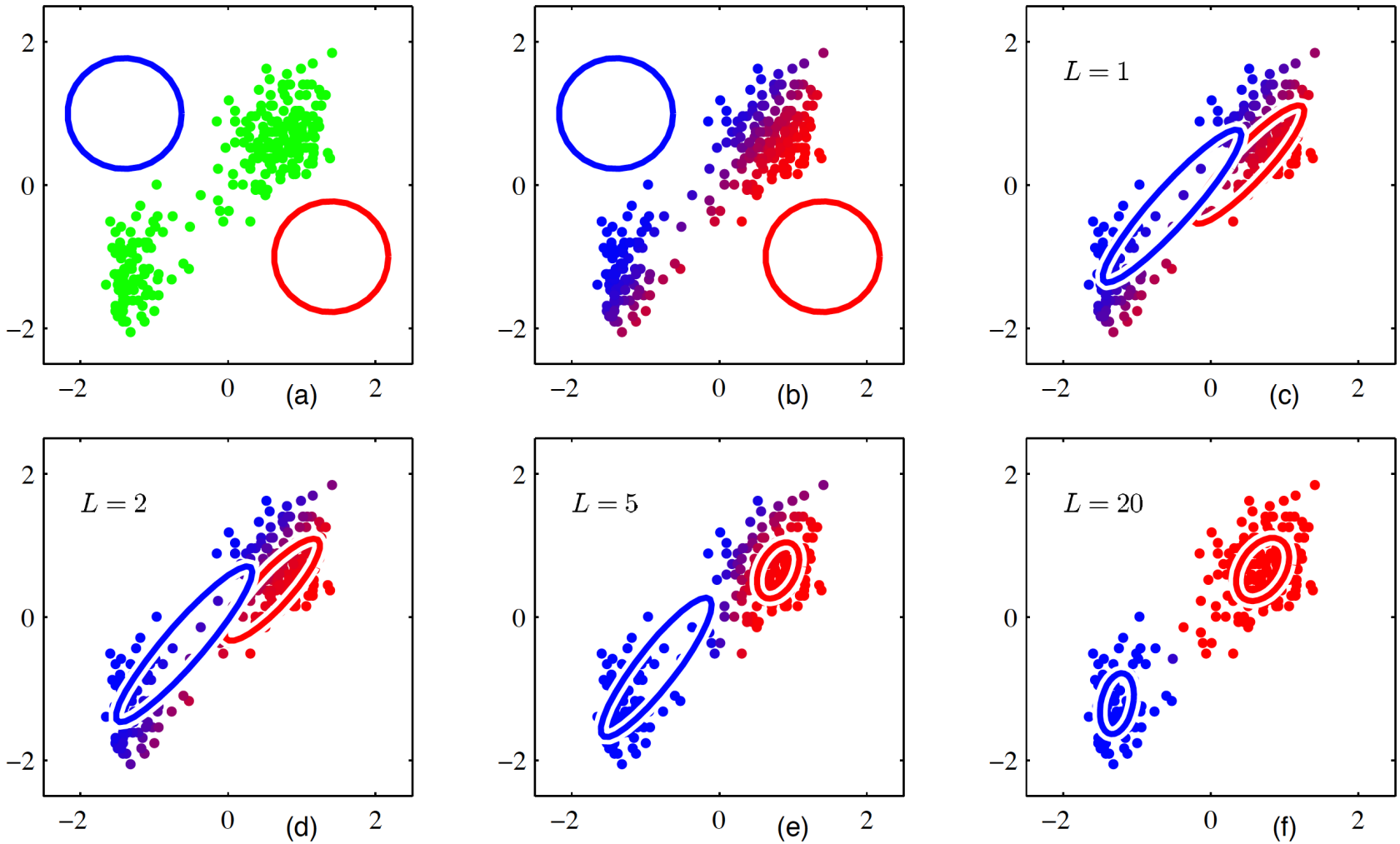
- Given iid samples $\mathcal{D} = \{x_1, ..., x_N\}$, the log likelihood that we need to maximize is

$$\begin{aligned} L(\theta) = \log p(\mathcal{D}|\theta) &= \log \prod_{n=1}^{N} p(x_n|\theta) \\ &= \sum_{n=1}^{N} \log p(x_n|\theta) \\ &= \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \end{aligned}$$

There is no Closed-form Solution

$$\sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)$$

- Can we compute $\frac{\partial L}{\partial \theta}$?
- Unfortunately, there is no closed-form solution that can find all parameters θ by a single computation.
- We will apply expectation-maximization (EM) algorithm.


2
0
−2
−2
0
2
(a)
(b)
$L = 1$
(c)
$L = 2$
(d)
$L = 5$
(e)
$L = 20$
(f)

Example: EM for isotropic GMM

Define

- $z_i \in [K]$: the cluster to which point x_i belongs.
- θ is a set of hyperparameters, i.e., $\theta = \{\{\pi_k, \mu_k, \sigma_k\}_{k=1}^K\}$
- $\mathcal{L}_c(\theta; \{x_i, z_i\}_{i\in[N]})$: the complete-data log-likelihood, i.e.,

$$\mathcal{L}_c(\theta; \{x_i, z_i\}_{i\in[N]}) = \sum_{i\in[N]} \log p(x_i, z_i \mid \theta)$$

Isotropic GMM

- Letting $z_{ik} = \mathbb{1}[z_i = k]$,

$$p(z_i) = \prod_{k\in[K]} \pi_k^{z_{ik}}, \quad \text{and} \quad p(x_i \mid z_i) = \prod_{k\in[K]} \left(\mathcal{N}(x_i \mid \mu_k, \sigma_k^2 I)\right)^{z_{ik}}.$$

Log-Likelihood for GMM (1)

The complete-data log-likelihood can be calculated as follows:

$$
\begin{aligned}
\mathcal{L}_c(\theta; \{x_i, z_i\}_{i\in[N]}) &= \sum_{i\in[N]} \log p(x_i, z_i \mid \theta) \\
&= \sum_{i\in[N]} \log \left(p(x_i \mid z_i, \theta) p(z_i \mid \theta)\right) \\
&= \sum_{i\in[N]} \log \left(\prod_{k\in[K]} \left(\pi_k \mathcal{N}(x_i \mid \mu_k, \sigma_k^2 I)\right)^{z_{ik}} \right) \\
&= \sum_{i\in[N]} \sum_{k\in[K]} z_{ik} \log \left(\pi_k \mathcal{N}(x_i \mid \mu_k, \sigma_k^2 I)\right) .
\end{aligned}
$$

Log-Likelihood for GMM (2)

For given θ', define the responsibility[1] r_{ik} of cluster k to data point x_i,

$$r_{ik} := p(z_i = k \mid x_i; \theta') = \mathbb{E}_{z_i|x_i,\theta'}[z_{ik}] \,.$$

Then, given θ' or $\{r_{ik}\}$, the marginal log-likelihood of $\theta = \{\mu_k, \sigma_k^2\}$ can be approximated by:

$$\begin{aligned}
\mathcal{Q}(\theta; \theta') &:= \mathbb{E}_{\{z_i\}_{i\in[N]}|\{x_i\}_{i\in[N]},\theta'} \left[\mathcal{L}_c(\theta; \{x_i, z_i\}_{i\in[N]})\right] \\
&= \sum_{i\in[N]} \sum_{k\in[K]} \mathbb{E}_{z_i|x_i,\theta'} \left[z_{ik} \log\left(\pi_k \mathcal{N}(x_i \mid \mu_k, \sigma_k^2 I)\right)\right] \\
&= \sum_{i\in[N]} \sum_{k\in[K]} r_{ik} \log\left(\pi_k \mathcal{N}(x_i \mid \mu_k, \sigma_k^2 I)\right) \\
&= \sum_{i\in[N]} \sum_{k\in[K]} r_{ik} \left[\log \pi_k - \frac{D}{2} \log \sigma_k^2 - \frac{1}{2\sigma_k^2} \|x_i - \mu_k\|^2\right] + \text{const} \,.
\end{aligned}$$

[1] A posterior of z given the other parameters.

EM for GMM

Starting from an arbitrary choice of $\theta = \left\{\pi_k, \mu_k, \sigma_k^2\right\}_{k\in[K]}$,

- E-step: Compute responsibilities $\{r_{ik}\}$ for given $\theta' = \theta$:

$$r_{ik} := p(z_i = k \mid x_i; \theta') = \frac{\pi_k p(x_i \mid z_i = k, \mu_k, \sigma_k^2)}{\sum_{\ell\in[K]} \pi_\ell p(x_i \mid z_i = \ell, \mu_\ell, \sigma_\ell^2)} \ .$$

- M-step: Update θ_{new} maximizing the approximated marginal log-likelihood $\mathcal{Q}(\theta; \theta')$:

$$\theta_{\text{new}} = \arg\max_{\theta} \ \mathcal{Q}(\theta; \theta') \ .$$

M-Step: Gaussian Parameters (1)

Using the theory of optimization, we find θ such that $\nabla_\theta \mathcal{Q}(\theta) = 0$:

- Mean

$$\frac{\partial \mathcal{Q}}{\partial \mu_k} = -\frac{1}{\sigma_k^2} \sum_{i \in [N]} r_{ik}(x_i - \mu_k) = 0$$

$$\implies \mu_{k,\text{new}} = \frac{\sum_{i \in [N]} r_{ik} x_i}{\sum_{i \in [N]} r_{ik}} \, .$$

- Variance

$$\frac{\partial \mathcal{Q}}{\partial \sigma_k^2} = \sum_{i \in [N]} r_{ik} \left[-\frac{D}{\sigma_k} + \frac{1}{\sigma_k^3} \|x_i - \mu_k\|^2 \right] = 0$$

$$\implies \sigma_{k,\text{new}}^2 = \frac{1}{D} \frac{\sum_{i \in [N]} r_{ik} \|x_i - \mu_{k,\text{new}}\|^2}{\sum_{i \in [N]} r_{ik}}$$

M-step: Mixing Parameter (2)

$$\mathcal{Q}(\theta) = \sum_{i\in[N]} \sum_{k\in[K]} r_{ik} \left[\log \pi_k - \frac{D}{2} \log \sigma_k^2 - \frac{1}{2\sigma_k^2} \|x_i - \mu_k\|^2 \right] + \text{const}$$

Note that $\{\pi_k\}$ must verify $\sum_{k\in[K]} \pi_k = 1$. Hence, recalling the theory of constrained optimization, consider the Lagrangian

$$\mathcal{Q}'(\theta, \lambda) = \mathcal{Q}(\theta) + \lambda \left(1 - \sum_{k\in[K]} \pi_k \right) .$$

Solving

$$\frac{\partial \mathcal{Q}'(\theta, \lambda)}{\partial \pi_k} = \sum_{i\in[N]} \frac{r_{ik}}{\pi_k} - \lambda = 0 ,$$

one can conclude that the optimal Lagrangian multiplier λ is given by $\lambda = N$, and thus

$$\pi_{k,\text{new}} = \frac{1}{N} \sum_{i\in[N]} r_{ik}$$

EM Algorithm for Isotropic GMM: Summary

Starting from an arbitrary choice of $\theta = \left\{\pi_k, \mu_k, \sigma_k^2\right\}_{k\in[K]}$,

- E-step: Compute responsibilities $\{r_{ik}\}$ for given $\theta' = \theta$:

$$r_{ik} := p(z_i = k \mid x_i; \theta') = \frac{\pi_k p(x_i \mid z_i = k, \mu_k, \sigma_k^2)}{\sum_{\ell\in[K]} \pi_\ell p(x_i \mid z_i = \ell, \mu_\ell, \sigma_\ell^2)} .$$

- M-step: Update θ_{new} maximizing the approximated log-likelihood:

$$\mu_{k,\text{new}} = \frac{\sum_{i\in[N]} r_{ik} x_i}{\sum_{i\in[N]} r_{ik}}$$

$$\sigma_{k,\text{new}}^2 = \frac{1}{D} \frac{\sum_{i\in[N]} r_{ik} \|x_i - \mu_{k,\text{new}}\|^2}{\sum_{i\in[N]} r_{ik}}$$

$$\pi_{k,\text{new}} = \frac{1}{N} \sum_{i\in[N]} r_{ik}$$

K-means: Special Case of EM for GMM?

Selecting $\pi_k = \frac{1}{K}$ and $\sigma_k^2 = \sigma^2$ for each $k \in [K]$ and infinitesimal $\sigma^2 \to 0$,

- E-step: Compute responsibilities $\{r_{ik}\}$ for given $\theta' = \theta$:

$$r_{ik} \to \mathbb{1}\left[k = \arg\max_{\ell \in [K]} p(x_i \mid z_i = \ell, \mu_\ell, \sigma^2)\right] .$$

- M-step: Update θ_{new} maximizing the approximated log-likelihood:

$$\mu_{k,\text{new}} = \frac{\sum_{i \in [N]} r_{ik} x_i}{\sum_{i \in [N]} r_{ik}}$$

$$\pi_{k,\text{new}} = \frac{1}{N} \sum_{i \in [N]} r_{ik}$$

Summary

Clustering: a problem of unsupervised learning

- Algorithms: K-means and Soft K-means
- Analysis: Gaussian Mixture Model (GMM) + expectation-maximization (EM)

What comes next: expectation-maximization (EM) and other unsupervised learning problems