CS570 Artificial Intelligence & Machine Learning

Mixture Models & Clustering

Kee-Eung Kim
Department of Computer Science
KAIST

Exponential Family Distributions

Exponential Family Distributions

- ☐ Gaussian, Bernoulli, Gamma, …

- ☐ Why important?
  - Can compress the data into a fixed-size summary without loss of information (Pitman-Koopman-Darmois theorem)
    => **online learning**
  - Only family of distributions for which conjugate priors exist
    => **easy to compute posterior**
  - Makes least set of assumptions except user-chosen constraints
    => **maximizes entropy**
  - useful for generalized linear models and variational inference

Definition of Exponential Family

$$p(\mathbf{x}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} h(\mathbf{x}) \exp[\boldsymbol{\theta}^\top \boldsymbol{\phi}(\mathbf{x})]$$
$$= h(\mathbf{x}) \exp[\boldsymbol{\theta}^\top \boldsymbol{\phi}(\mathbf{x}) - A(\boldsymbol{\theta})]$$

- $Z(\boldsymbol{\theta}) = \int_{\mathcal{X}} h(\mathbf{x}) \exp[\boldsymbol{\theta}^\top \boldsymbol{\phi}(\mathbf{x})] d\mathbf{x}$
- $A(\boldsymbol{\theta}) = \log Z(\boldsymbol{\theta})$

- $\boldsymbol{\theta}$: natural parameters (canonical parameters)
- $\boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^d$: sufficient statistics
- $h(\mathbf{x})$: scaling constant, often 1
- $Z(\boldsymbol{\theta})$: partition function
- $A(\boldsymbol{\theta})$: log partition function (cumulant function)

- "natural" exponential family: $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$

Example: Bernoulli and Gaussian

- Bernoulli distribution

$$\mathrm{Ber}(x|\mu) = \mu^x(1-\mu)^{1-x} = \exp[x\log\mu + (1-x)\log(1-\mu)]$$
$$= \exp[\boldsymbol{\theta}^\top \boldsymbol{\phi}(x)]$$

where $\boldsymbol{\phi}(x) = [\mathbb{I}(x=0), \mathbb{I}(x=1)]$ and $\boldsymbol{\theta} = [\log\mu, \log(1-\mu)]$

  - Alternatively:

$$\mathrm{Ber}(x|\mu) = (1-\mu)\exp[x\log(\tfrac{\mu}{1-\mu})]$$

so that $\phi(x) = x$ and $\theta = \log(\frac{\mu}{1-\mu})$

- Univariate Gaussian distribution

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp[-\tfrac{1}{2\sigma^2}(x-\mu)^2]$$
$$= \frac{1}{\sqrt{2\pi\sigma^2}}\exp[-\tfrac{1}{2\sigma^2}x^2 + \tfrac{\mu}{\sigma^2}x - \tfrac{1}{2\sigma^2}\mu^2]$$
$$= \frac{1}{Z(\boldsymbol{\theta})}\exp[\boldsymbol{\theta}^\top \boldsymbol{\phi}(x)]$$

where $\boldsymbol{\theta} = [\mu/\sigma^2, \frac{-1}{2\sigma^2}]$, $\boldsymbol{\phi}(x) = [x, x^2]$, $Z(\mu,\sigma^2) = \sqrt{2\pi\sigma^2}\exp[\frac{\mu^2}{2\sigma^2}]$

Log Partition Function

- Why $A(\boldsymbol{\theta})$ is called cumulant function
  - first derivative

$$\frac{dA}{d\theta} = \frac{d}{d\theta}(\log \int \exp(\theta\phi(x))h(x)dx)$$
$$= \frac{\frac{d}{d\theta}\int \exp(\theta\phi(x))h(x)dx}{\int \exp(\theta\phi(x))h(x)dx} = \frac{\int \phi(x)\exp(\theta\phi(x))h(x)dx}{\exp A(\theta)}$$
$$= \int \phi(x)\exp(\theta\phi(x) - A(\theta))h(x)dx$$
$$= \int \phi(x)p(x)dx = E[\phi(x)]$$

  - second derivative

$$\frac{d^2A}{d\theta^2} = \int \phi(x)\exp(\theta\phi(x) - A(\theta))h(x)(\phi(x) - A'(\theta))dx$$
$$= \int \phi(x)p(x)(\phi(x) - A'(\theta))dx$$
$$= \int \phi^2(x)p(x)dx - A'(\theta)\int \phi(x)p(x)dx$$
$$= E[\phi^2(X)] - E[\phi(x)]^2 = \mathrm{Var}[\phi(x)]$$

  - first and second derivatives generates cumulants of sufficient statistics

MLE for Exponential Family

- likelihood:

$$p(\mathcal{D}|\boldsymbol{\theta}) = \left[\prod_{i=1}^{N} h(\mathbf{x}_i)\right] \frac{1}{Z(\boldsymbol{\theta})^N} \exp\left[\boldsymbol{\theta}^\top \boldsymbol{\phi}(\mathcal{D})\right]$$

  - where $\boldsymbol{\phi}(\mathcal{D}) = [\sum_{i=1}^N \phi_1(\mathbf{x}_i), \ldots, \sum_{i=1}^N \phi_K(\mathbf{x}_i)]$
- log-likelihood: $\log p(\mathcal{D}|\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{\phi}(\mathcal{D}) - NA(\boldsymbol{\theta}) + C$
  - concave since $-A(\boldsymbol{\theta})$ is concave (why?)
    => unique global maximum
- MLE: $\nabla_{\boldsymbol{\theta}} \log p(\mathcal{D}|\boldsymbol{\theta}) = \boldsymbol{\phi}(\mathcal{D}) - N \cdot E[\boldsymbol{\phi}(X)] = 0$
  - $E[\boldsymbol{\phi}(X)] = \frac{1}{N}\sum_{i=1}^N \boldsymbol{\phi}(\mathbf{x}_i)$
  - moment matching: empirical average of sufficient statistics must equal the model's theoretical expected sufficient statistics
  - e.g. Bernoulli: $E[\boldsymbol{\phi}(X)] = p(X=1) = \hat{\mu}_{\text{MLE}} = \frac{1}{N}\sum_{i=1}^N \mathbb{I}(x_i = 1)$

Mixture Models

Semiparametric Density Estimation

- Parametric
  - Assume a single model $p(\mathbf{x}|\boldsymbol{\theta})$
- Nonparametric
  - No model; data speaks for itself
- Semiparametric
  - $p(\mathbf{x}|\boldsymbol{\theta})$ is a mixture of densities
  - Multiple possible explanations/prototypes
    - Different handwriting styles
    - Accents in speech

Mixture Models

$$p(\mathbf{x}_i) = \sum_{k=1}^{K} p(z_i) p_k(\mathbf{x}_i|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p_k(\mathbf{x}_i|\boldsymbol{\theta})$$

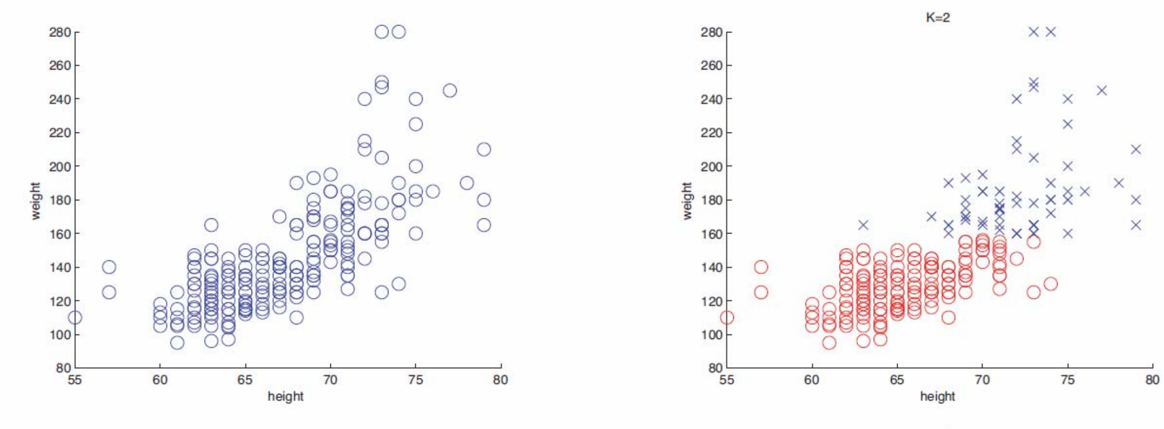
where $z_i \in \{1, \ldots, K\}$ are the latent state/groups/clusters

$p(z_i) = \mathrm{Cat}(\boldsymbol{\pi})$ mixture proportions (priors)

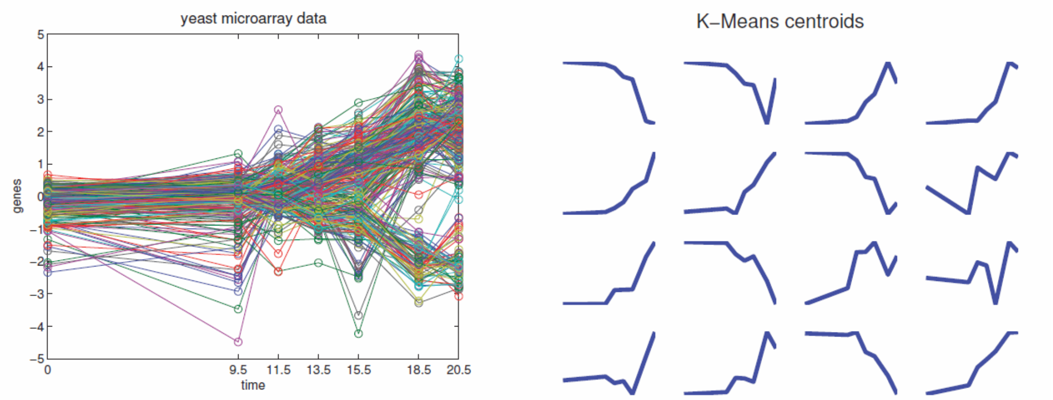
$p_k(\mathbf{x}_i|\boldsymbol{\theta})$ component densities

- In parametric classification, we had $z_i = y_i$
  - We knew which instance belongs to which group (i.e. given as labels in supervised learning)
- Special case: Gaussian mixture when $p_k(\mathbf{x}_i|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
  - Parameters: $\boldsymbol{\theta} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$
  - Unlabeled sample $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{N}$

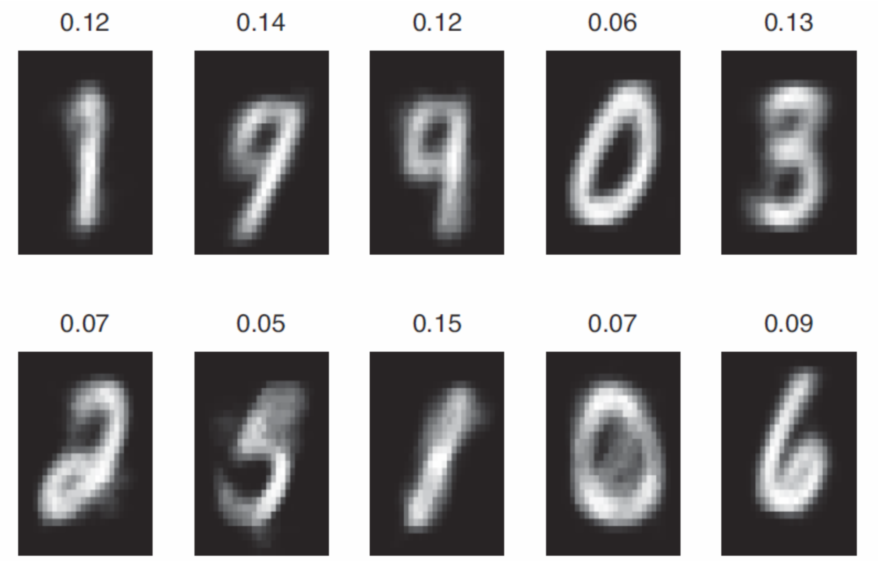
Mixture Models for Clustering



(a) (b)



(a) (b)



Classes vs. Clusters

- Supervised: $\mathcal{D} = \{\mathbf{x}_i, y_i\}$
- Classes $y_i, \quad i = 1, \ldots, K$
- Density:

$$p(\mathbf{x}_i) = \sum_{k=1}^{K} p(y_i = k) p_k(\mathbf{x}_i | \boldsymbol{\theta})$$

where $p_k(\mathbf{x}_i|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

- MAP estimation:

$$\hat{p}(y_i = k) = \frac{\sum_i \mathbb{I}(y_i{=}k)}{N}$$

$$\hat{\boldsymbol{\mu}}_k = \frac{\sum_i \mathbb{I}(y_i{=}k)\mathbf{x}_i}{\sum_i \mathbb{I}(y_i{=}k)}$$

$$\hat{\boldsymbol{\Sigma}}_k = \frac{\sum_i \mathbb{I}(y_i{=}k)(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^\top}{\sum_i \mathbb{I}(y_i{=}k)}$$

- Unsupervised: $\mathcal{D} = \{\mathbf{x}_i\}$
- Clusters $z_i, \quad i = 1, \ldots, K$
- Density:

$$p(\mathbf{x}_i) = \sum_{k=1}^{K} p(z_i = k) p_k(\mathbf{x}_i | \boldsymbol{\theta})$$

where $p_k(\mathbf{x}_i|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

- MAP estimation:

Labels z_i ??

Non-Convexity in ML/MAP Estimation

- ☐ log-likelihood for a latent variable model (LVM)

$$\log p(\mathcal{D}|\boldsymbol{\theta}) = \sum_i \log \left[\sum_{\mathbf{z}_i} p(\mathbf{x}_i, \mathbf{z}_i|\boldsymbol{\theta})\right]$$

- ☐ and assume an exponential family distribution

$$p(\mathbf{x}_i, \mathbf{z}_i|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp[\boldsymbol{\theta}^\top \boldsymbol{\phi}(\mathbf{x}_i, \mathbf{z}_i)]$$

- ☐ **Complete data log likelihood**:

$$\ell_c(\boldsymbol{\theta}) = \sum_i \log p(\mathbf{x}_i, \mathbf{z}_i|\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \left(\sum_i \boldsymbol{\phi}(\mathbf{x}_i, \mathbf{z}_i)\right) - NA(\boldsymbol{\theta})$$

  - linear function – convex function = concave function

- ☐ **Observed data log likelihood**:

$$\ell(\boldsymbol{\theta}) = \sum_i \log \sum_{\mathbf{z}_i} p(\mathbf{x}_i, \mathbf{z}_i|\boldsymbol{\theta}) = \sum_i \log \sum_{\mathbf{z}_i} \left[e^{\boldsymbol{\theta}^\top \boldsymbol{\phi}(\mathbf{x}_i, \mathbf{z}_i)}\right] - NA(\boldsymbol{\theta})$$

  - convex function – convex function = ??

- ☐ Practical approach: local optimizer with multiple random restarts

A Naive Optimization Approach

- Idea: use a generic gradient-based optimizer to find a local minimum of the negative log likelihood (NLL)

$$\mathrm{NLL}(\boldsymbol{\theta}) \equiv -\frac{1}{N} \log p(\mathcal{D}|\boldsymbol{\theta})$$

- Can you see some problems with this approach?

Expectation-Maximization Algorithm

Expectation-Maximization

- ☐ Since complete data log likelihood is not available, use the **expected complete data log likelihood** (a.k.a auxiliary function)

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}) \equiv E[\ell_c(\boldsymbol{\theta})|\mathcal{D}, \boldsymbol{\theta}^{t-1}]$$

- ☐ Iterate the two steps (ML estimation)
  - E-step: compute $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1})$
    - The terms inside Q which the MLE depends on
  - M-step: find $\boldsymbol{\theta}^t = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1})$

- ☐ For MAP estimation, change the M-step to

$$\boldsymbol{\theta}^t = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}) + \log p(\boldsymbol{\theta})$$

EM for GMMs (1)

- expected complete data log likelihood (auxiliary function):

$$
\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}) \equiv E[\ell_c(\boldsymbol{\theta}) | \mathcal{D}, \boldsymbol{\theta}^{t-1}] &= E[\textstyle\sum_i \log p(\mathbf{x}_i, z_i | \boldsymbol{\theta})] \\
&= \textstyle\sum_i E\left[\log\left[\prod_{k=1}^K (\pi_k p(\mathbf{x}_i | \boldsymbol{\theta}_k))^{\mathbb{I}(z_i = k)}\right]\right] \\
&= \textstyle\sum_i \sum_k E[\mathbb{I}(z_i = k)] \log[\pi_k p(\mathbf{x}_i | \boldsymbol{\theta}_k)] \\
&= \textstyle\sum_i \sum_k p(z_i = k | \mathbf{x}_i, \boldsymbol{\theta}^{t-1}) \log[\pi_k p(\mathbf{x}_i | \boldsymbol{\theta}_k)] \\
&= \textstyle\sum_i \sum_k r_{ik} \log \pi_k + \sum_i \sum_k r_{ik} \log p(\mathbf{x}_i | \boldsymbol{\theta}_k)
\end{aligned}
$$

  where $r_{ik} \equiv p(z_i = k | \mathbf{x}_i, \boldsymbol{\theta}^{t-1})$ is the responsibility of the k-th cluster for $\mathbf{x}_i$

- E-step: compute r_{ik}
- M-step: find $\boldsymbol{\theta}^t = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1})$

EM for GMMs (2)

- expected complete data log likelihood (auxiliary function):

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}) = \sum_i \sum_k r_{ik} \log \pi_k + \sum_i \sum_k r_{ik} \log p(\mathbf{x}_i | \boldsymbol{\theta}_k)$$

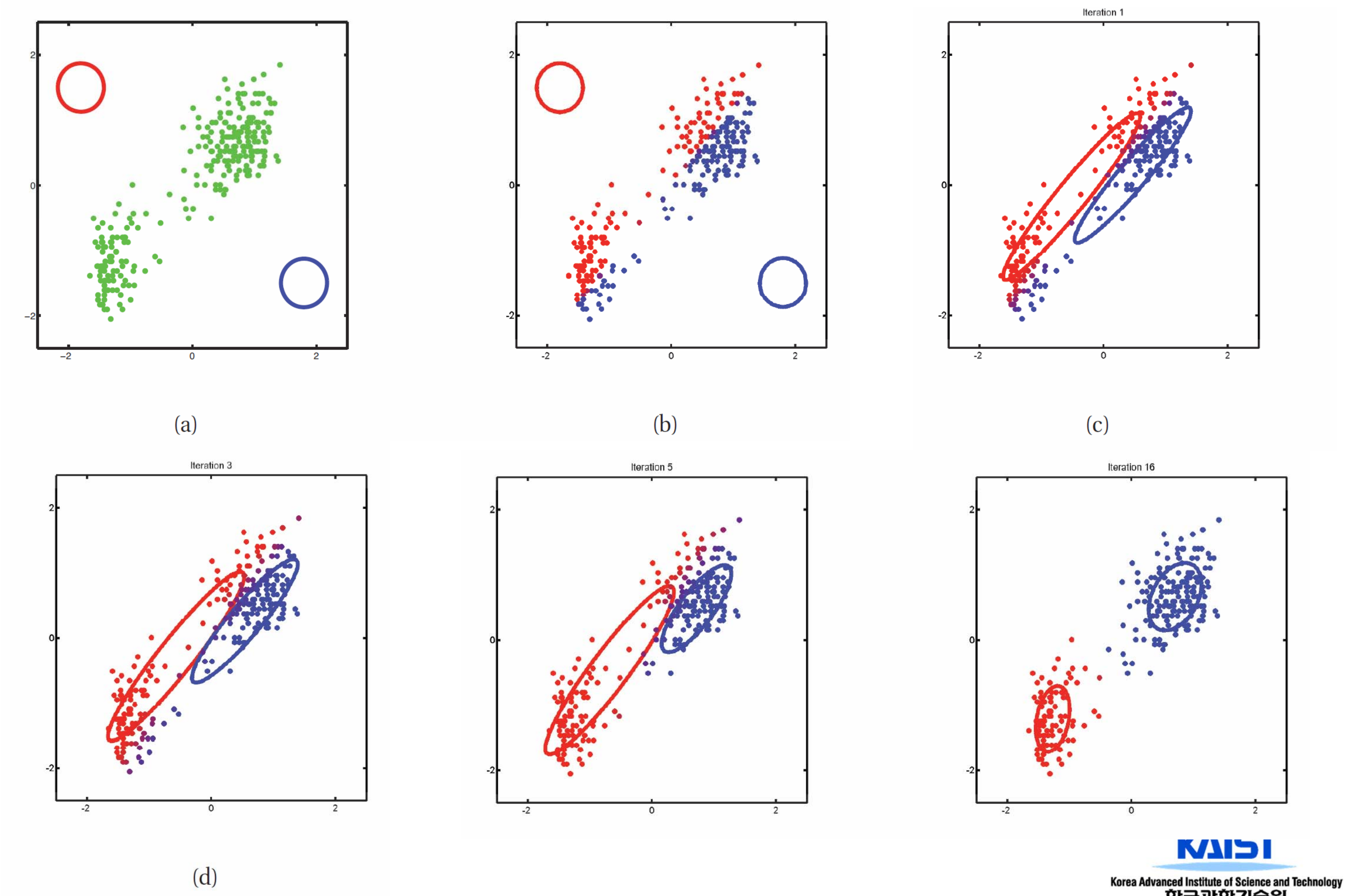
  where $r_{ik} \equiv p(z_i = k | \mathbf{x}_i, \boldsymbol{\theta}^{t-1})$ is the responsibility of the k-th cluster for $\mathbf{x}_i$

- E-step:
  - same for any mixture model: $r_{ik} = \dfrac{\pi_k p(\mathbf{x}_i | \boldsymbol{\theta}_k^{t-1})}{\sum_{k'} \pi_{k'} p(\mathbf{x}_i | \boldsymbol{\theta}_{k'}^{t-1})}$
- M-step:
  - From $\max\limits_{\pi_k} \sum_i \sum_k r_{ik} \log \pi_k$: $\pi_k = \frac{1}{N} \sum_i r_{ik}$
  - From $\max\limits_{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k} \sum_i \sum_k r_{ik} \log p(\mathbf{x}_i, \boldsymbol{\theta}_k)$

$$= -\frac{1}{2} \sum_i r_{ik} [\log |\boldsymbol{\Sigma}_k| + (\mathbf{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k)]$$

$$\boldsymbol{\mu}_k = \frac{\sum_i r_{ik} \mathbf{x}_i}{\sum_i r_{ik}}, \quad \boldsymbol{\Sigma}_k = \frac{\sum_i r_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^\top}{\sum_i r_{ik}} = \frac{\sum_i r_{ik} \mathbf{x}_i \mathbf{x}_i^\top}{\sum_i r_{ik}} - \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top$$

EM for GMMs



GMM in Practice...

- The risk of overfitting in Gaussian mixtures
  - Individual covariance matrices when high dimensional input & few samples

- Possible solutions
  - Assume a common covariance matrix
  - Assume a diagonal form for individual covariance matrices

  - Dimensionality reduction for each cluster (advanced topic to be discussed later)

Clustering for Supervised Learning

- What does the unsupervised learning do?
  - Clustering finds similarities between instances
    - N instances is reduced to k groups
  - Dimensionality reduction* finds correlations between variables
    - d-dimensional data is reduced to k-dimensional data

- Use #1: After clustering
  - Easier for human to analyze and label the data, using some visualization
- Use #2: clustering as preprocessing for supervised learning
  - Estimated group labels $0 \leq z_i \leq 1$ may be seen as the elements for a new k-dimensional space, where we can learn discriminant or regressor
  - However, k could be set larger than d if appropriate

Mixture of Mixtures

- In classification, the input comes from a mixture of classes (supervised)
- If each class is also a mixture, e.g., of Gaussians (unsupervised), we have a mixture of mixtures

$$p(\mathbf{x}_i|y_i = c) = \sum_{j=1}^{K_c} p(\mathbf{x}_i|z_{jk})p(z_{ji})$$

$$p(\mathbf{x}_i) = \sum_{c=1}^{C} p(\mathbf{x}_i|y_i = c)p(c)$$

Choosing the Number of Clusters

- In some applications, k is clearly defined by the requirement
  - Color quantization
- Plot data in 2D using PCA, and check for obvious clusters
- Incremental approach
  - Try one more cluster at a time until “elbow” of reconstruction error/log likelihood/intergroup distances
- Manual inspection
  - Experts check whether clusters actually represent something meaningful
- Dirichlet Process Mixture Model (DPMM)