[2018 Elice Machine Learning Basic Course]

Introduction to K-Means Clustering and Gaussian Mixture Models

Heung-Il Suk

hisuk@korea.ac.kr

http://www.ku-milab.org

Department of Brain and Cognitive Engineering,
Korea University

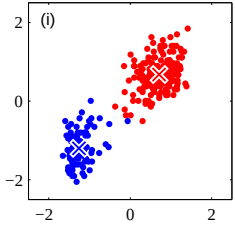
June 5, 2018

Contents

K-Means Clustering

K-means Clustering

- Given a data set $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ in a D-dimensional Euclidean space
- Goal: to partition the data set into some number K of clusters
  - Intuitively, comprising a group of data points whose inter-point distances are small compared with the distances to points outside of the cluster



[Problem definition]

- Prototype $\boldsymbol{\mu}_k$ associated with the k-th cluster; center of the cluster
- Binary indicator variable $r_{nk} \in \{0, 1\}$
  - which of the K clusters the data point $\mathbf{x}_n$ is assigned to
  - known as the '*1-of-K coding scheme*'

- Objective: to find (1) an assignment of data points to clusters $\{r_{nk}\}$ as well as (2) a set of vectors $\{\boldsymbol{\mu}_k\}$, such that the sum of the squares of the distances of each data point to its closest vector $\boldsymbol{\mu}_k$, is a minimum.

$$\min_{\{r_{nk}\},\{\boldsymbol{\mu}_k\}} J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \left\| \mathbf{x}_n - \boldsymbol{\mu}_k \right\|^2$$

K-means algorithm

Iterative and successive optimizations with respect to the $\{r_{nk}\}$ and the $\{\boldsymbol{\mu}_k\}$

1. Choose some initial values for the $\{\boldsymbol{\mu}_k\}$
2. Minimize J w.r.t. the $\{r_{nk}\}$, keeping the $\{\boldsymbol{\mu}_k\}$ fixed (expectation)
   - Estimating the expected cluster
3. Minimize J w.r.t. the $\{\boldsymbol{\mu}_k\}$, keeping the $\{r_{nk}\}$ fixed (maximization)
   - Maximizing the likelihood
4. Repeat this two-stage optimization until convergence

K-means algorithm

Iterative and successive optimizations with respect to the $\{r_{nk}\}$ and the $\{\boldsymbol{\mu}_k\}$

1. Choose some initial values for the $\{\boldsymbol{\mu}_k\}$
2. Minimize J w.r.t. the $\{r_{nk}\}$, keeping the $\{\boldsymbol{\mu}_k\}$ fixed (**expectation**)
   - Estimating the expected cluster
3. Minimize J w.r.t. the $\{\boldsymbol{\mu}_k\}$, keeping the $\{r_{nk}\}$ fixed (**maximization**)
   - Maximizing the likelihood
4. Repeat this two-stage optimization until convergence

Determination of the $\{r_{nk}\}$ (expectation)

- J: a linear function of $\{r_{nk}\}$
  - Optimization: a closed form solution

- Independence among terms involving different n,
  - Optimization for each n separately
  - By choosing r_{nk} to be 1 for whichever value of k gives the minimum value of $\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$

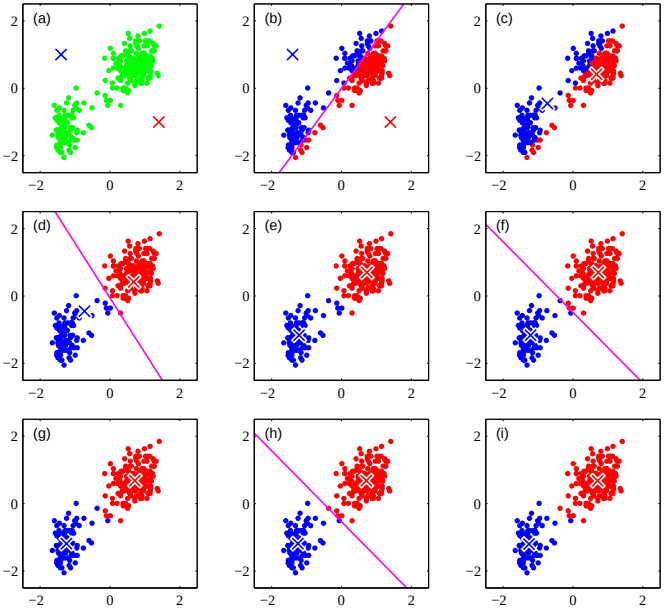
$$r_{nk} = \begin{cases} 1 & \text{if } k = \underset{j}{\operatorname{argmin}} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 \\ 0 & \text{otherwise} \end{cases}$$

Optimization of the $\{\boldsymbol{\mu}_k\}$ (maximization)

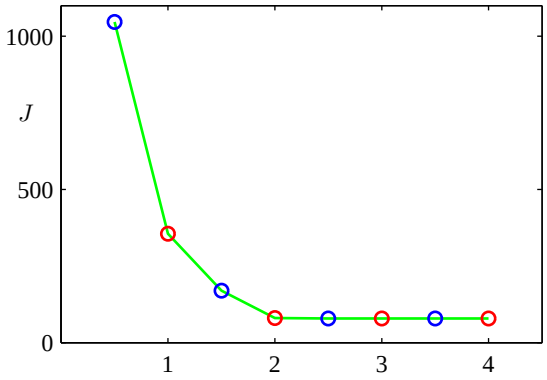
- J: a quadratic function of $\boldsymbol{\mu}_k$.
- Optimization: by setting its derivative w.r.t. $\boldsymbol{\mu}_k$ to zero

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}$$

  - $\boldsymbol{\mu}_k$: mean of all of the data points $\mathbf{x}_n$ assigned to cluster k


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)

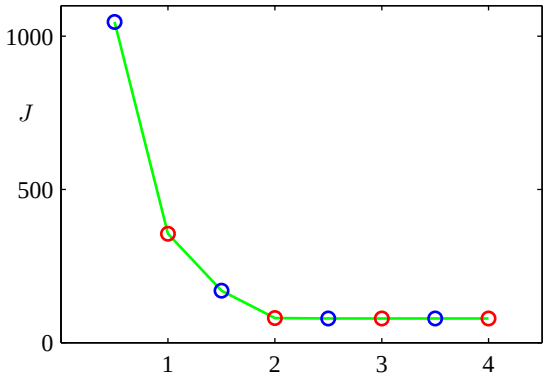
Plot of the cost function J after each step



assignment (expectation) step, updating (maximization) step

Non-decreasing after each stage in iterations

Plot of the cost function J after each step



assignment (expectation) step, updating (maximization) step

Non-decreasing after each stage in iterations

- Poor initial values for cluster centers
  - Several steps are involved for convergence (illustration purpose)
- Better initialization is to chose μ_k to be a random subset of K data points
- K-means algorithm itself is often used to initialize the parameters in a Gaussian mixture model before applying the EM algorithm

- Poor initial values for cluster centers
  - Several steps are involved for convergence (illustration purpose)
- Better initialization is to chose $\boldsymbol{\mu}_k$ to be a random subset of K data points
- K-means algorithm itself is often used to initialize the parameters in a Gaussian mixture model before applying the EM algorithm

- Poor initial values for cluster centers
  - Several steps are involved for convergence (illustration purpose)
- Better initialization is to chose $\boldsymbol{\mu}_k$ to be a random subset of K data points
- K-means algorithm itself is often used to initialize the parameters in a Gaussian mixture model before applying the EM algorithm

Implementation of the K-means algorithm

- Direct implementation can be relatively slow
  - in each expectation step, need to compute the Euclidean distance between every prototype vector and every data point

$$\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

- Speeding up
  - Precomputing a data structure (*e.g.*, tree) such that nearby points are in the same subtree (Ramasubramanian and Paliwal, 1990; Moore, 2000)
  - Making use of the triangle inequality for distances, thereby avoiding unnecessary distance calculations (Hodgson, 1998; Elkan, 2003)

Online stochastic algorithm [MacQueen, 1967]

- By applying the Robbins-Monro procedure (Chapter 2.3.4) to the problem of finding the roots of the regression function

$$\min_{\{r_{nk}\},\{\boldsymbol{\mu}_k\}} J = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

given by the derivatives of J w.r.t. $\boldsymbol{\mu}_k$

$$\boldsymbol{\mu}_k^{\text{new}} = \boldsymbol{\mu}_k^{\text{old}} + \eta_n \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{old}}\right)$$

  - η_n: learning rate parameter, typically made to decrease monotonically as more data points are considered

▸ Robbins-Monro algorithm

Dissimilarity measure

- K-means: based on the use of squared Euclidean distance
- Limit the type of data variables
  - Inappropriate for cases where some or all of the variables represent categorical labels for instance
  - Making the determination of the cluster means non-robust to outliers

- More general dissimilarity measure $\mathcal{V}(\mathbf{x}, \mathbf{x}')$

$$\tilde{J} = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \mathcal{V}(\mathbf{x}, \boldsymbol{\mu}_k)$$

- K-*medoids* algorithm [Kaufman and Rousseeuw, 1987]
  - Maximization step: potentially more complex than for K-means, so it is common to restrict each cluster prototype to be equal to one of the data vectors assigned to that cluster

Dissimilarity measure

- K-means: based on the use of squared Euclidean distance
- Limit the type of data variables
  - Inappropriate for cases where some or all of the variables represent categorical labels for instance
  - Making the determination of the cluster means non-robust to outliers

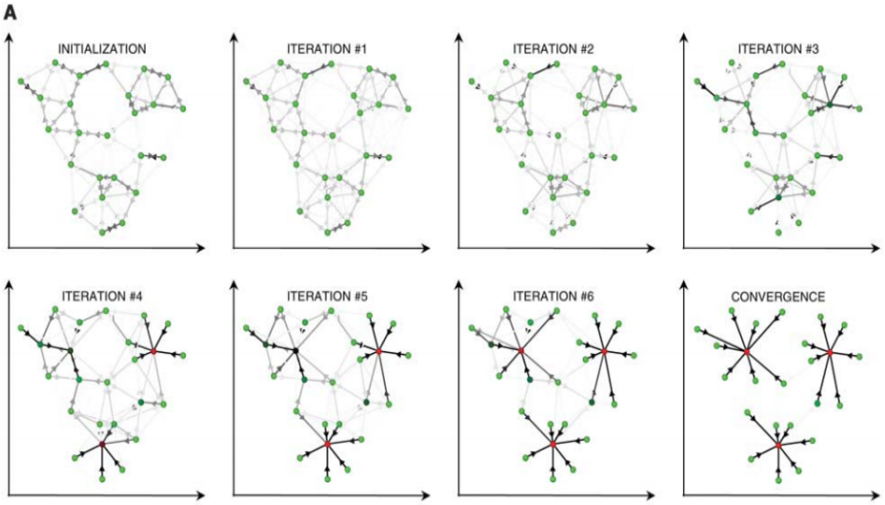
- More general dissimilarity measure $\mathcal{V}(\mathbf{x}, \mathbf{x}')$

$$\tilde{J} = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \mathcal{V}(\mathbf{x}, \boldsymbol{\mu}_k)$$

- K-*medoids* algorithm [Kaufman and Rousseeuw, 1987]
  - Maximization step: potentially more complex than for K-means, so it is common to restrict each cluster prototype to be equal to one of the data vectors assigned to that cluster

Affinity Propagation [Frey and Dueck, Science 2007)]

- Not require the number of clusters to be determined or estimated before running the algorithm

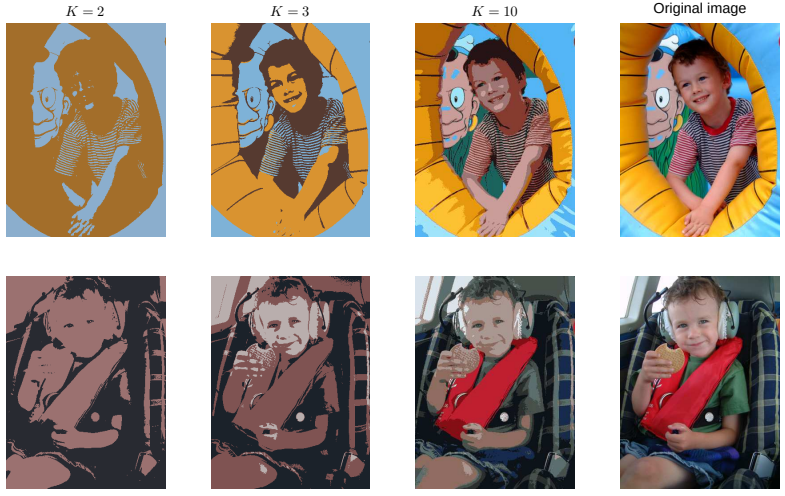


Application of K-Means

Image segmentation

- Partition an image into regions
  - each of which has homogeneous visual appearance
  - or corresponds to objects
  - or parts of objects

- Each pixel is a point in $[R, G, B]$ space
- K-means clustering is used with a palette of K colors
- Methods does not take into account proximity of different pixels.


$K = 2$
$K = 3$
$K = 10$
Original image

Lossy data compression

- Accept some errors in the reconstruction in return for higher levels of compression than can be achieved in the lossless case
- For each of the N data points, we store only the identity k of the cluster to which it is assigned.
- Also store the values of the K cluster centers $\boldsymbol{\mu}_k$, where $K \ll N$
- known as *vector quantization*; $\{\boldsymbol{\mu}_k\}$ called *code-book vectors*

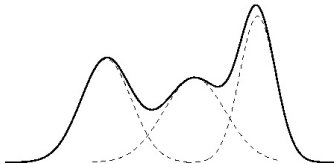
Limitation of K-means

- Hard assignment: every data point is assigned uniquely to one and only one cluster
- A point may lie roughly midway between cluster centers.
- A *probabilistic approach* will have a 'soft' assignment of data points to clusters in a way that reflects the level of uncertainty over the most appropriate assignment

Mixtures of Gaussians

Introduction

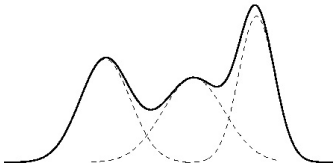
- Introduction of latent variables allows complicated distributions to be formed from simpler components
  - Mixture distributions (*e.g.*, Gaussian mixture): discrete latent variables
  - Continuous latent variables (Chapter 12)



- Mixture models
  - provide a framework for building more complex probability distributions
  - used to cluster data (*clustering*)

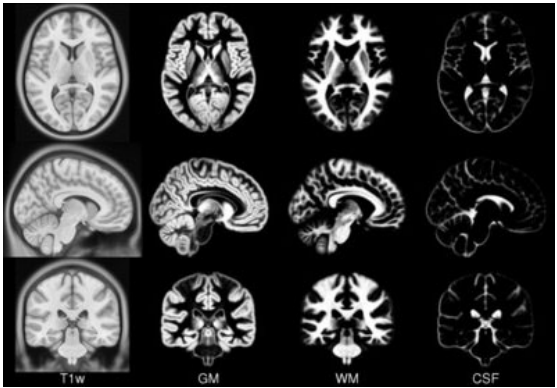
Introduction

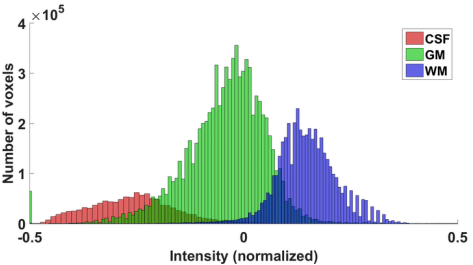
- Introduction of latent variables allows complicated distributions to be formed from simpler components
  - Mixture distributions (*e.g.*, Gaussian mixture): discrete latent variables
  - Continuous latent variables (Chapter 12)



- Mixture models
  - provide a framework for building more complex probability distributions
  - used to cluster data (*clustering*)

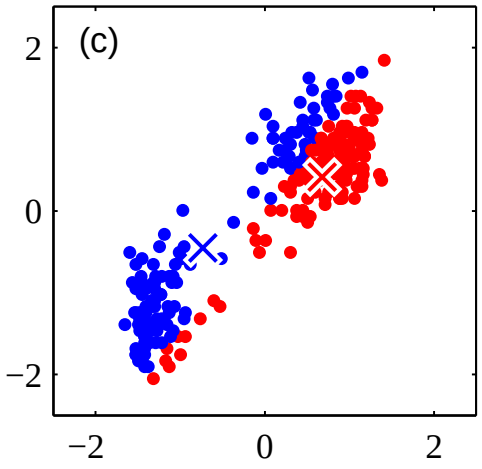
Brain tissue segmentation

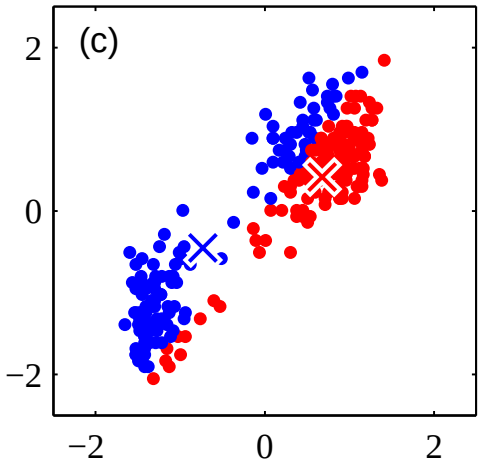


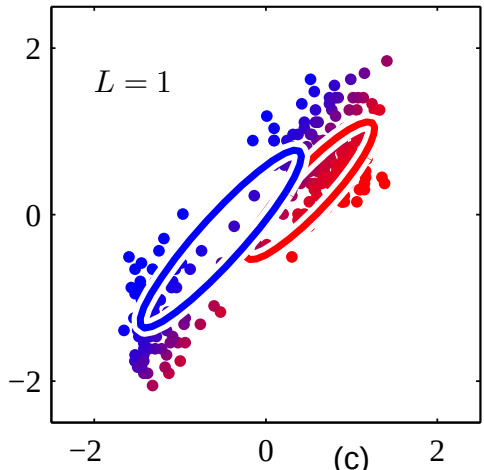


- K-means algorithm: non-probabilistic technique
  - Identifying groups, or clusters, of data points in a multidimensional space
- Latent variable view of mixture distributions
  - Discrete latent variables: interpreted as defining *assignments* of data points to specific components of the mixture
- Expectation-Maximization (EM) algorithm
  - MLE-based approach
  - Bayesian treatment in the framework of *variational inference*: requiring little computation compared with EM, and resolving the principal difficulties of ML while also allowing the number of components in the mixture to be inferred automatically from the data

- K-means algorithm: non-probabilistic technique
  - Identifying groups, or clusters, of data points in a multidimensional space
- Latent variable view of mixture distributions
  - Discrete latent variables: interpreted as defining *assignments* of data points to specific components of the mixture
- Expectation-Maximization (EM) algorithm
  - MLE-based approach
  - Bayesian treatment in the framework of *variational inference*: requiring little computation compared with EM, and resolving the principal difficulties of ML while also allowing the number of components in the mixture to be inferred automatically from the data


(c)
2
0
−2
−2
0
2


(c)
2
0
−2
−2
0
2

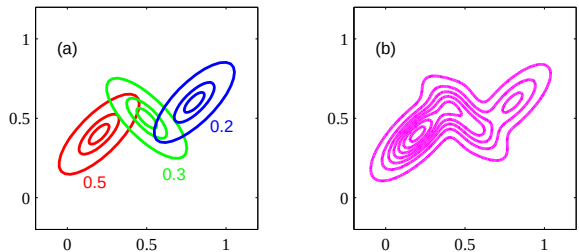

$L = 1$
2
0
−2
−2
0
2
(c)

Gaussian Mixture Model (GMM)

A simple linear superposition of Gaussian components

- Providing a richer class of density models than the single Gaussian

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k\right)$$



Formulation of Gaussian mixtures in terms of discrete *latent* variables.

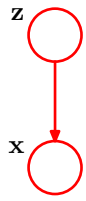
- Introduce a K-dimensional binary random variable $\mathbf{z}$ having a 1-of-K coding scheme
  - a particular element z_k equal to 1 and all the other elements equal to 0

$$\sum_{k=1}^{K} z_k = 1$$

- K possible states for the vector $\mathbf{z}$ according to which element is nonzero

- Define the joint distribution

$$p(\mathbf{x}, \mathbf{z}) = \underbrace{p(\mathbf{x}|\mathbf{z})}_{\text{conditional}} \underbrace{p(\mathbf{z})}_{\text{marginal}}$$



Graphical representation of a mixture model

Probabilistic Graphical Models

- Marginal distribution over $\mathbf{z}$: $p(\mathbf{z})$
  - Specified in terms of the mixing coefficients $\{\pi_k\}$

$$p(z_k = 1) = \pi_k \qquad \text{where } 0 \leq \pi_k \leq 1, \sum_{k=1}^{K} \pi_k = 1$$

  - Due to a 1-of-K representation

$$p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}$$

- Conditional distribution of $\mathbf{x}$ given a particular value for $\mathbf{z}$

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)$$

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)^{z_k}$$

- Marginal distribution over $\mathbf{z}$: $p(\mathbf{z})$
  - Specified in terms of the mixing coefficients $\{\pi_k\}$

$$p(z_k = 1) = \pi_k \qquad \text{where } 0 \leq \pi_k \leq 1, \sum_{k=1}^{K} \pi_k = 1$$

  - Due to a 1-of-K representation

$$p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}$$

- Conditional distribution of $\mathbf{x}$ given a particular value for $\mathbf{z}$

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k\right)$$

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k\right)^{z_k}$$

- Marginal distribution of $\mathbf{x}$: $p(\mathbf{x})$

$$
\begin{aligned}
p(\mathbf{x}) &= \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) \\
&= \sum_{\mathbf{z}} p(\mathbf{z}) p(\mathbf{x}|\mathbf{z}) \\
&= \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k\right)
\end{aligned}
$$

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z})$$

- Given several observations $\mathbf{x}_1, \ldots, \mathbf{x}_N$, for every observed data point $\mathbf{x}_n$, a corresponding latent variable $\mathbf{z}_n$

- Instead of the marginal distribution $p(\mathbf{x})$, **we work with the joint distribution** $p(\mathbf{x}, \mathbf{z})$
  - Leading to significant simplifications
  - Most notably through the introduction of the Expectation-Maximization (EM) algorithm

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z})$$

- Given several observations $\mathbf{x}_1, \ldots, \mathbf{x}_N$, for every observed data point $\mathbf{x}_n$, a corresponding latent variable $\mathbf{z}_n$
- Instead of the marginal distribution $p(\mathbf{x})$, **we work with the joint distribution** $p(\mathbf{x}, \mathbf{z})$
  - Leading to significant simplifications
  - Most notably through the introduction of the Expectation-Maximization (EM) algorithm

- Conditional probability of $\mathbf{z}$ given $\mathbf{x}$

$$
\begin{aligned}
p(z_k = 1|\mathbf{x}) &= \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)} \\
&= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \Sigma_j)} \equiv \gamma(z_k)
\end{aligned}
$$

- View π_k as the prior probability of $z_k = 1$
- $\gamma(z_k)$ as the posterior probability once we have observed $\mathbf{x}$
  - also viewed as the *responsibility* that component k takes for 'explaining' the observation $\mathbf{x}$

- Conditional probability of $\mathbf{z}$ given $\mathbf{x}$

$$
\begin{aligned}
p(z_k = 1|\mathbf{x}) &= \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)} \\
&= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \Sigma_j)} \equiv \gamma(z_k)
\end{aligned}
$$

- View π_k as the prior probability of $z_k = 1$
- $\gamma(z_k)$ as the posterior probability once we have observed $\mathbf{x}$
  - also viewed as the *responsibility* that component k takes for 'explaining' the observation $\mathbf{x}$

- Conditional probability of $\mathbf{z}$ given $\mathbf{x}$

$$
\begin{aligned}
p(z_k = 1|\mathbf{x}) &= \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)} \\
&= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \Sigma_j)} \equiv \gamma(z_k)
\end{aligned}
$$

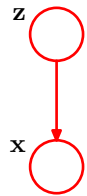
- View π_k as the prior probability of $z_k = 1$
- $\gamma(z_k)$ as the posterior probability once we have observed $\mathbf{x}$
  - also viewed as the *responsibility* that component k takes for 'explaining' the observation $\mathbf{x}$

- Conditional probability of $\mathbf{z}$ given $\mathbf{x}$

$$
\begin{aligned}
p(z_k = 1|\mathbf{x}) &= \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\mathbf{x}|z_j = 1)} \\
&= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \Sigma_j)} \equiv \gamma(z_k)
\end{aligned}
$$

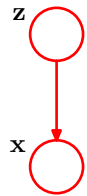
- View π_k as the prior probability of $z_k = 1$
- $\gamma(z_k)$ as the posterior probability once we have observed $\mathbf{x}$
  - also viewed as the *responsibility* that component k takes for 'explaining' the observation $\mathbf{x}$

Ancestral sampling to generate random samples distributed according to the Gaussian mixture model

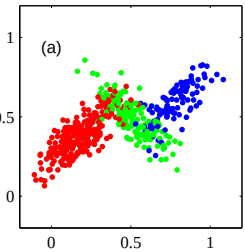


1. Generate a value for $\mathbf{z}$, denoted as $\hat{\mathbf{z}}$, from the marginal distribution $p(\mathbf{z})$
2. Generate a value for $\mathbf{x}$ from the conditional distribution $p(\mathbf{x}|\hat{\mathbf{z}})$

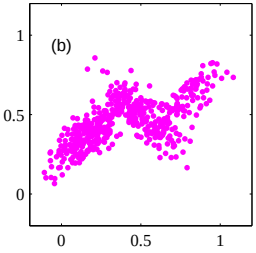
Ancestral sampling to generate random samples distributed according to the Gaussian mixture model



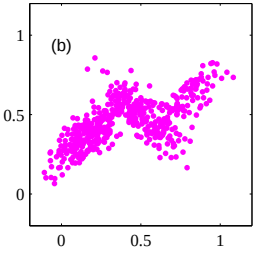
1. Generate a value for $\mathbf{z}$, denoted as $\hat{\mathbf{z}}$, from the marginal distribution $p(\mathbf{z})$
2. Generate a value for $\mathbf{x}$ from the conditional distribution $p(\mathbf{x}|\hat{\mathbf{z}})$



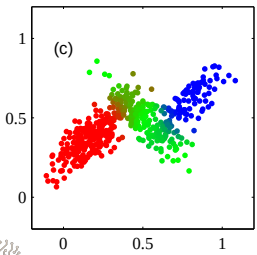
(a) Samples from $p(\mathbf{x}, \mathbf{z})$ are plotted according to value of $\mathbf{x}$ and colored with value of $\mathbf{z}$



(b) Samples from marginal distribution $p(\mathbf{x})$ obtained by ignoring values of $\mathbf{z}$



(b) Samples from marginal distribution $p(\mathbf{x})$ obtained by ignoring values of $\mathbf{z}$



(c) Representing the value of the responsibilities $\gamma(z_{nk})$ associated with data point $\mathbf{x}_n$ by plotting the corresponding point using proportions of red, blue, and green ink given by $\gamma(z_{nk})$ for $k = 1, 2, 3$, respectively

Machine Intelligence Lab

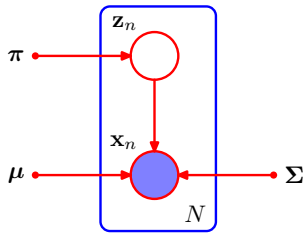
Maximum Likelihood for GMM

Given a set of N i.i.d. observations $\{\mathbf{x}_n\}_{n=1}^N$, model this data using a mixture of Gaussians

$$\mathbf{X} = \left[\mathbf{x}_1^\top; \cdots; \mathbf{x}_N^\top\right] \in \mathbb{R}^{N \times D}$$

$$\mathbf{Z} = \left[\mathbf{z}_1^\top; \cdots; \mathbf{z}_N^\top\right] \in \mathbb{R}^{N \times K}$$

$$\mathbf{z}_n \in \{0,1\}^K, \quad \sum_{k=1}^K z_{nk} = 1$$



Goal: to estimate the three sets of parameters

$$\{\pi_k\}_{k=1}^K, \quad \{\boldsymbol{\mu}_k\}_{k=1}^K, \quad \{\Sigma_k\}_{k=1}^K$$

Likelihood function

$$p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \underbrace{\left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k) \right\}}_{p(\mathbf{x}_n|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}$$

Log of the likelihood function

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k) \right\}$$

Goal: to estimate the three sets of parameters

$$\{\pi_k\}_{k=1}^K, \quad \{\boldsymbol{\mu}_k\}_{k=1}^K, \quad \{\Sigma_k\}_{k=1}^K$$

Likelihood function

$$p\left(\mathbf{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}\right) = \prod_{n=1}^{N} \underbrace{\left\{\sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_k,\Sigma_k\right)\right\}}_{p(\mathbf{x}_n|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma})}$$

Log of the likelihood function

$$\ln p\left(\mathbf{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}\right) = \sum_{n=1}^{N} \ln \left\{\sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_k,\Sigma_k\right)\right\}$$

Goal: to estimate the three sets of parameters

$$\{\pi_k\}_{k=1}^K, \quad \{\boldsymbol{\mu}_k\}_{k=1}^K, \quad \{\Sigma_k\}_{k=1}^K$$

Likelihood function

$$p\left(\mathbf{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}\right) = \prod_{n=1}^{N} \underbrace{\left\{\sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_k,\Sigma_k\right)\right\}}_{p(\mathbf{x}_n|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma})}$$

Log of the likelihood function

$$\ln p\left(\mathbf{X}|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}\right) = \sum_{n=1}^{N} \ln \left\{\sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_k,\Sigma_k\right)\right\}$$

- A more complex problem than for the case of a single Gaussian
  - Presence of the summation over k that appears inside the logarithm
  - No longer obtain a closed form solution

- Iterative approaches
  - Gradient-based optimization techniques (Fletcher, 1987; Nocedal and Wright, 1999; Bishop and Nabney, 2008) (c.f., mixture density network)
  - **EM algorithm**
    - Broad applicability
    - Foundations for a discussion of variational inference techniques

Two significant problems associated with the maximum likelihood framework applied to Gaussian mixture models

- Presence of singularities
- Problem of identifiability [Casellan and Berger, 2002]

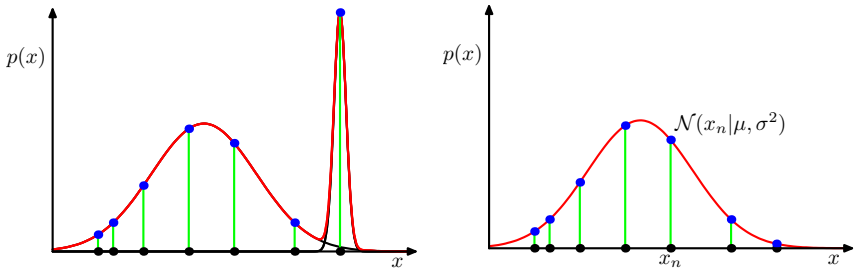
Singularities with Gaussian mixtures

- Consider a Gaussian mixture
  - Components have covariance matrices $\Sigma_k = \sigma_k^2 \mathbf{I}$
- Suppose that the j-th component has its mean $\boldsymbol{\mu}_j$ exactly equal to one of the data points $\mathbf{x}_n$

$$\mathcal{N}\left(\mathbf{x}_n | \mathbf{x}_n, \sigma_j^2 \mathbf{I}\right) = \frac{1}{(2\pi)^{1/2}} \frac{1}{\sigma_j}$$

- $\sigma_j \to 0 \quad \Rightarrow \quad \mathcal{N}\left(\mathbf{x}_n | \mathbf{x}_n, \sigma_j^2 \mathbf{I}\right) \to \infty$

- Thus, the maximization of the log likelihood function is not well-posed.
  - Such singularities will occur whenever one of the Gaussian components 'collapses' onto a specific data point.
  - Not arise in the case of a single Gaussian distribution



- If a single Gaussian collapses onto a data point, it will contribute multiplicative factors to the likelihood function arising from the other data points and these factors will go to zero exponentially fast, giving an overall likelihood that goes to zero rather than infinity.
- However, once we have (at least) two components in the mixture, one of the components can have a finite variance and therefore assign finite probability to all of the data points while the other component can shrink onto one specific data point and thereby contribute an ever increasing additive value to the log likelihood. (*overfitting*)
- This difficulty does not occur if we adopt a *Bayesian* approach.

- For the moment, however, we simply note that in applying maximum likelihood to Gaussian mixture models we must take steps to avoid finding such pathological solutions and instead seek local maxima of the likelihood function that are well behaved.

- We can hope to avoid the singularities by using suitable heuristics, for instance by detecting when a Gaussian component is collapsing and resetting its mean to a randomly chosen value while also resetting its covariance to some large value, and then continuing with the optimization.

Problem of identifiability

- For any given maximum likelihood solution, a K-component mixture will have a total of $K!$ equivalent solutions
  - corresponding to the $K!$ ways of assigning K sets of parameters to K components
- In other words, for any given (nondegenerate) point in the space of parameter values there will be a further $K! - 1$ additional points all of which give rise to exactly the same distribution.
  - An important issue when wish to interpret the parameter values discovered by a model
  - Also arising for continuous latent variables
- However, for the purposes of finding a good density model, it is irrelevant because any of the equivalent solutions is as good as any other.

EM for Gaussian Mixtures

- Expectation-Maximization (EM) algorithm
  - a method for finding *maximum likelihood* solutions for models with latent variables [Dempster *et al.*, 1977; McLachlan and Krishnan, 1997]
  - Broad applicability in the context of a variety of different models
- In the context of the Gaussian mixture model

$$\ln p\left(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k\right) \right\}$$

- Taking derivatives w.r.t. $\boldsymbol{\mu}_k$, Σ_k, π_k and setting to zero

$$\max_{\boldsymbol{\mu}_k} \ln p\left(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

- Taking derivatives w.r.t. $\boldsymbol{\mu}_k$ and setting to zero

$$\begin{aligned} 0 &= \sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_j, \Sigma_j\right)}}_{p(z_k=1|\mathbf{x}_n) \equiv \gamma(z_{nk})} \Sigma_k^{-1}\left(\mathbf{x}_n - \boldsymbol{\mu}_k\right) \\ &= \sum_{n=1}^{N} \gamma(z_{nk}) \Sigma_k^{-1}\left(\mathbf{x}_n - \boldsymbol{\mu}_k\right) \end{aligned}$$

- Multiplying both sides by Σ_k (assuming non-singular)

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n$$

where $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$: effective number of points assigned to cluster k

$$\max_{\boldsymbol{\mu}_k} \ln p\left(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

- Taking derivatives w.r.t. $\boldsymbol{\mu}_k$ and setting to zero

$$\begin{aligned} 0 &= \sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_j, \Sigma_j\right)}}_{p(z_k=1|\mathbf{x}_n) \equiv \gamma(z_{nk})} \Sigma_k^{-1}\left(\mathbf{x}_n - \boldsymbol{\mu}_k\right) \\ &= \sum_{n=1}^{N} \gamma(z_{nk}) \, \Sigma_k^{-1}\left(\mathbf{x}_n - \boldsymbol{\mu}_k\right) \end{aligned}$$

- Multiplying both sides by Σ_k (assuming non-singular)

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \, \mathbf{x}_n$$

where $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$: effective number of points assigned to cluster k

$$\max_{\Sigma_k} \ln p\left(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

- Taking derivatives w.r.t. Σ_k and setting to zero
  - Making use of the result for the maximum likelihood solution for the covariance matrix of a single Gaussian

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \left(\mathbf{x}_n - \boldsymbol{\mu}_k\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_k\right)^\top$$

where $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$: effective number of points assigned to cluster k

$$\max_{\pi_k} \ln p\left(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \text{ s.t. } \sum_{k=1}^{K} \pi_k = 1$$

- Constrained optimization: Lagrangian multiplier

$$\ln p\left(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

- Taking derivatives w.r.t. π_k and setting to zero

$$0 = \sum_{n=1}^{N} \frac{\mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_j, \Sigma_j\right)} + \lambda$$

- Multiplying both sides by π_k and sum over k: $\lambda = -N$

$$\pi_k = \frac{\gamma(z_{nk})}{N} = \frac{N_k}{N}$$

- Mixing coefficient for the k-th component is given by the average responsibility which that component takes for explaining the data points

- The results for $\boldsymbol{\mu}_k$, Σ_k, and π_k are not a closed-form solution
  - $\because$ responsibilities $\gamma(z_{nk})$ depend on those parameters in a complex way

$$\frac{\pi_k \mathcal{N}\left(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\mathbf{x}_n | \boldsymbol{\mu}_j, \Sigma_j\right)}$$

- However, the results suggest a simple *iterative scheme* for finding a solution to the maximum likelihood problem.

EM algorithm for GMM

1. Choose some initial values for the means $\{\boldsymbol{\mu}_k\}$, covariances $\{\Sigma_k\}$, and mixing coefficients $\{\pi_k\}$
2. Alternate between the following two updates until convergence
   - (E-step) Use the current values for the parameters to evaluate the posterior probabilities, or responsibilities

$$p(z_k = 1|\mathbf{x}_n) = \gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \Sigma_j)}$$

   - (M-step) Use these probabilities to re-estimate the means, covariances, and mixing coefficients

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})\, \mathbf{x}_n$$

$$\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)^{\top}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N} \quad \text{where } N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

EM algorithm for GMM

1. Choose some initial values for the means $\{\boldsymbol{\mu}_k\}$, covariances $\{\Sigma_k\}$, and mixing coefficients $\{\pi_k\}$
2. Alternate between the following two updates until convergence
   - (**E-step**) Use the current values for the parameters to **evaluate the posterior probabilities, or responsibilities**

$$p(z_k = 1|\mathbf{x}_n) = \gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \Sigma_j)}$$

   - (**M-step**) Use these probabilities to **re-estimate the means, covariances, and mixing coefficients**

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n$$

$$\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)^{\top}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N} \quad \text{where } N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

EM algorithm for GMM

1. Choose some initial values for the means $\{\boldsymbol{\mu}_k\}$, covariances $\{\Sigma_k\}$, and mixing coefficients $\{\pi_k\}$
2. Alternate between the following two updates until convergence
   - (**E-step**) Use the current values for the parameters to **evaluate the posterior probabilities, or responsibilities**

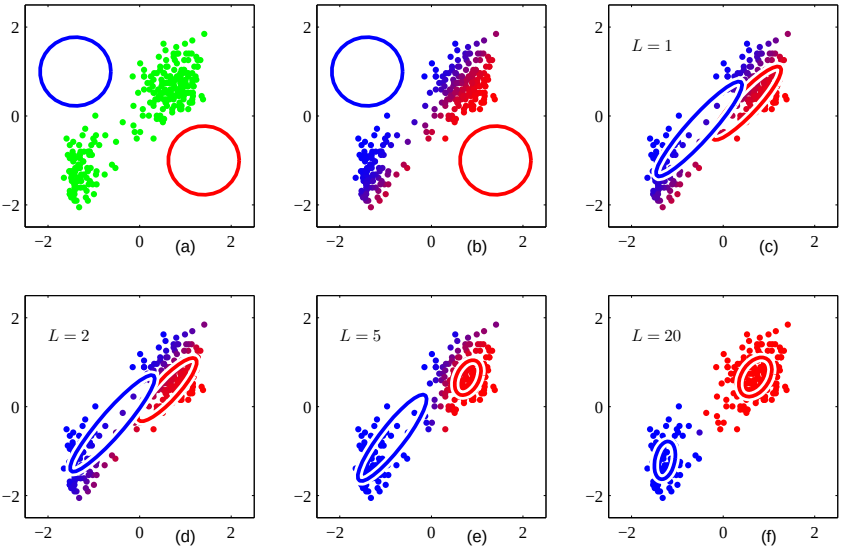
$$p(z_k = 1|\mathbf{x}_n) = \gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \Sigma_j)}$$

   - (**M-step**) Use these probabilities to **re-estimate the means, covariances, and mixing coefficients**

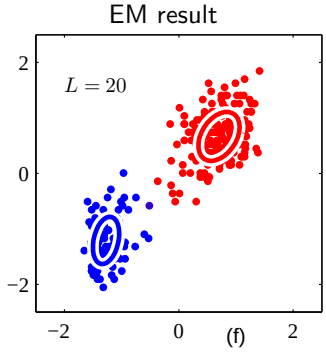
$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n$$

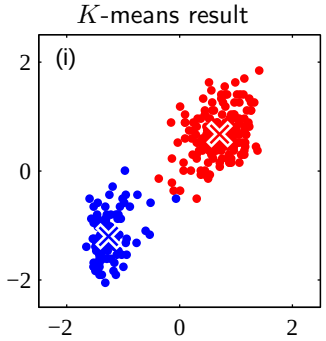
$$\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)^{\top}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N} \quad \text{where } N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$



Machine Intelligence Lab





- EM takes many more iterations to reach convergence.
- Each cycle requires significantly more computation.
- Common to run K-means first in order to find a suitable initialization
  - Covariance matrices: sample covariances of the clusters
  - Mixing coefficients: fractions of data points assigned to the respective clusters
- EM is not guaranteed to find the global maximum of the log likelihood function.

Thank you
for your attention!!!

(Q & A)

hisuk (AT) korea.ac.kr

http://www.ku-milab.org

Machine Intelligence Lab

Supplementary

Robbins-Monro algorithm

- Defines a sequence of successive estimates of the root θ^*

$$\theta^{(N)} = \theta^{(N-1)} - a_{N-1} z\left(\theta^{(N-1)}\right)$$

  - $z\left(\theta^{(N)}\right)$: an observed value of z when θ takes the value $\left(\theta^{(N)}\right)$
  - Coefficients $\{a_N\}$: a sequence of positive numbers satisfying

$$\lim_{N\to\infty} a_N = 0, \quad \sum_{N=1}^{\infty} a_N = \infty, \quad \sum_{N=1}^{\infty} a_N^2 < \infty$$

- A general maximum likelihood problem can be solved sequentially using the Robbins-Monro algorithm

Probabilistic Graphical Models (PGM)

Framework for representing dependencies among the random variables

Probability Theory

Graph Theory

Probability Theory

- Sum rule

$$p(A) = \int p(A, B)\, dB$$

- Product (chain) rule

$$p(A, B) = p(B|A)\, p(A)$$

- Bayes rule

$$p(B|A) = \frac{p(A|B)\, p(B)}{p(A)}$$

(Statistical) Independence

- Marginal independence

$$A \perp\!\!\!\perp B \equiv p(A, B) = p(A)\, p(B)$$

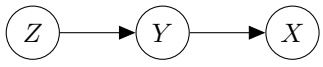
- Conditional independence

$$\begin{aligned} A \perp\!\!\!\perp B|C &\equiv p(A, B|C) = p(A|C)\, p(B|C) \\ &\equiv p(A|B, C) = p(A|C) \end{aligned}$$

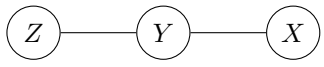
Graph theory: $\mathbb{G}(V, E)$

Graph Terminology

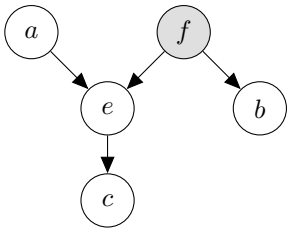
- V (vertices or nodes): represents random variables
  - Observed measurements, parameters, latent variables, hypothesis
- E (edges or links): represents *probabilistic* relationships between variables

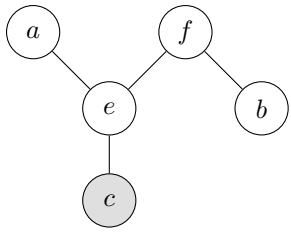


causal relation

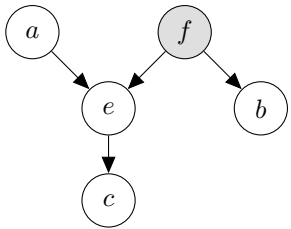


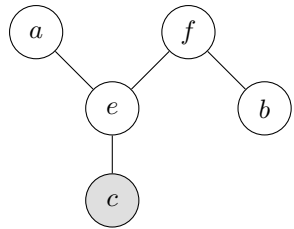
corrleation





- The graph captures the way in which the *joint distribution* over all of the random variables can be *decomposed into a product of factors* each depending only on a subset of the variables.





- The graph captures the way in which the *joint distribution* over all of the random variables can be *decomposed into a product of factors* each depending only on a subset of the variables.

Useful properties of PGM

- A simple way to visualize the structure of a probabilistic model
- Used to design and motivate new models
- Insights into the properties of the model, including conditional independence properties
- Complex computations can be expressed in terms of graphical manipulation, in which underlying mathematical expressions are carried along implicitly.

- The pattern of edges in the graph represents the qualitative dependencies between the variables; the absence of an edge between two nodes means that any statistical dependency between these two variables is mediated via some other variable or set of variables.

- The quantitative dependencies between variables which are connected via edges are specified via parameterized conditional distributions, or more generally non-negative potential functions. The pattern of edges and the potential functions together specify a joint probability distribution over all the variables in the graph.

- **Bayesian Networks (BN)**
  - *Directed* Acyclic Graph (DAG)
  - Useful for expressing causal relationships between random variables
  - A graphical way to represent a particular factorization of a joint distribution
- **Markov Random Fields (MRF)**
  - *Undirected* Graphical Models
  - The links do not carry arrows and have no directional significance
  - Better suited to expressing soft constrains between random variables
- **Chain Graphs**: including both directed and undirected links