732A99/TDDE01 Machine Learning

Lecture 1b Block 2: Mixture Models

Jose M. Peña
IDA, Linköping University, Sweden

Contents

Literature

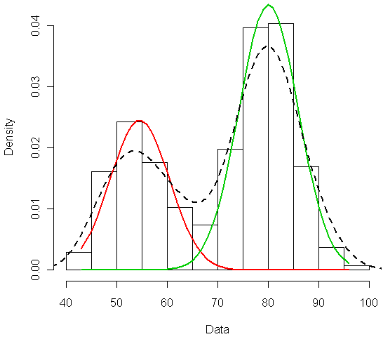
- Main source
  - Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006. Sections 2.3.9, 9.1-9.3.3 and 14.5.3.
- Additional source
  - Hastie, T., Tibshirani, R. and Friedman, J. *The Elements of Statistical Learning*. Springer, 2009. Section 8.5.

Mixture Models

- Sometimes the data do not follow any known probability distribution but a mixture of known distributions such as

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} p(k)p(\boldsymbol{x}|k)$$

where $p(\boldsymbol{x}|k)$ are called mixture components and $p(k)$ are called mixing coefficients, which are usually denoted by π_k and $0 \leq \pi_k \leq 1$ and $\sum_k \pi_k = 1$.

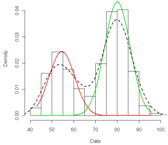


Mixture Models

- Sometimes the data do not follow any known probability distribution but a mixture of known distributions such as

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} p(k)p(\boldsymbol{x}|k)$$

  where $p(\boldsymbol{x}|k)$ are called mixture components and $p(k)$ are called mixing coefficients, which are usually denoted by π_k and $0 \leq \pi_k \leq 1$ and $\sum_k \pi_k = 1$.



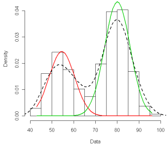
- We can also see a mixture model as an ensemble model of a population with subpopulations:
  1. Choose a subpopulation according to $Multinomial(k|\pi_1, \ldots, \pi_K)$.
  2. Sample an instance from the chosen subpopulation according to $p(\boldsymbol{x}|k)$.

Mixture Models

- Sometimes the data do not follow any known probability distribution but a mixture of known distributions such as

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} p(k)p(\boldsymbol{x}|k)$$

where $p(\boldsymbol{x}|k)$ are called mixture components and $p(k)$ are called mixing coefficients, which are usually denoted by π_k and $0 \leq \pi_k \leq 1$ and $\sum_k \pi_k = 1$.

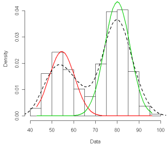


- We can also see a mixture model as an ensemble model of a population with subpopulations:
  1. Choose a subpopulation according to $Multinomial(k|\pi_1, \ldots, \pi_K)$.
  2. Sample an instance from the chosen subpopulation according to $p(\boldsymbol{x}|k)$.
- Mixture of multivariate Gaussian distributions:

$$p(\boldsymbol{x}) = \sum_k \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \text{ and } \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{2\pi^{D/2}} \frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_k)}$$

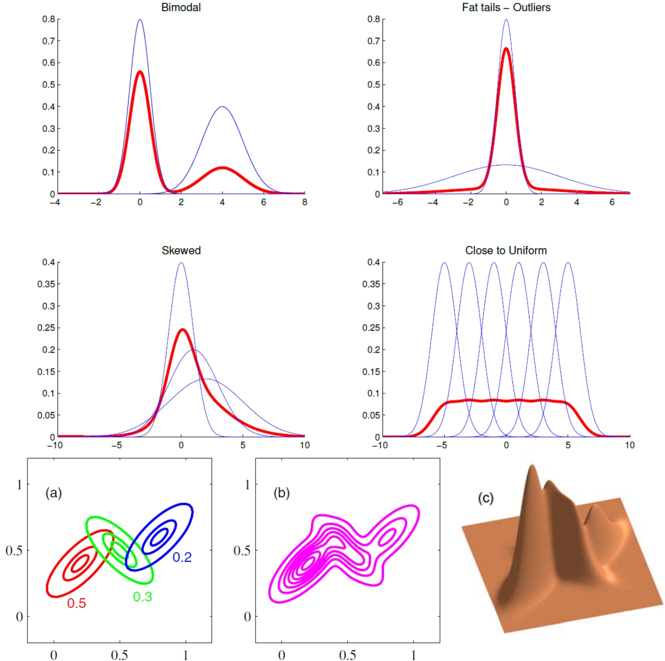
Mixture Models

- Sometimes the data do not follow any known probability distribution but a mixture of known distributions such as
$$p(\boldsymbol{x}) = \sum_{k=1}^{K} p(k)p(\boldsymbol{x}|k)$$
where $p(\boldsymbol{x}|k)$ are called mixture components and $p(k)$ are called mixing coefficients, which are usually denoted by π_k and $0 \leq \pi_k \leq 1$ and $\sum_k \pi_k = 1$.



- We can also see a mixture model as an ensemble model of a population with subpopulations:
  1. Choose a subpopulation according to $\mathit{Multinomial}(k|\pi_1, \ldots, \pi_K)$.
  2. Sample an instance from the chosen subpopulation according to $p(\boldsymbol{x}|k)$.
- Mixture of multivariate Gaussian distributions:
$$p(\boldsymbol{x}) = \sum_{k} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \text{ and } \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{2\pi^{D/2}} \frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)}$$
- Note that a mixture model defines a proper probability distribution:
$$0 \leq p(\boldsymbol{x}) \leq 1 \text{ and } \int p(\boldsymbol{x}) d\boldsymbol{x} = 1$$

Mixture Models



Mixture Models

- Mixture of multivariate Bernoulli distributions:

$$p(\boldsymbol{x}) = \sum_k \pi_k Bernoulli(\boldsymbol{x}|\boldsymbol{\mu}_k)$$

where we assume that

$$Bernoulli(\boldsymbol{x}|\boldsymbol{\mu}_k) = \prod_i Bernoulli(x_i|\mu_{ki}) = \prod_i \mu_{ki}^{x_i}(1-\mu_{ki})^{(1-x_i)}$$

Mixture Models

- Mixture of multivariate Bernoulli distributions:

$$p(\boldsymbol{x}) = \sum_k \pi_k Bernoulli(\boldsymbol{x}|\boldsymbol{\mu}_k)$$

where we assume that

$$Bernoulli(\boldsymbol{x}|\boldsymbol{\mu}_k) = \prod_i Bernoulli(x_i|\mu_{ki}) = \prod_i \mu_{ki}^{x_i}(1-\mu_{ki})^{(1-x_i)}$$

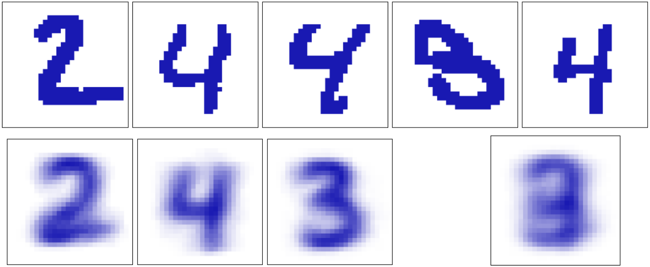


Figure 9.10 Illustration of the Bernoulli mixture model in which the top row shows examples from the digits data set after converting the pixel values from grey scale to binary using a threshold of 0.5. On the bottom row the first three images show the parameters μ_{ki} for each of the three components in the mixture model. As a comparison, we also fit the same data set using a single multivariate Bernoulli distribution, again using maximum likelihood. This amounts to simply averaging the counts in each pixel and is shown by the right-most image on the bottom row.

Maximum Likelihood

- Given a sample $\{\boldsymbol{x}_n, k_n\}$ of size N from a mixture of multivariate Bernoulli distributions, rewrite it as $\{\boldsymbol{x}_n, \boldsymbol{z}_n\}$ where $\boldsymbol{z}_n$ is a K-dimensional binary vector having only the k_n-th element equal to 1.

[1] Any stationary point of the Lagrangian function is a stationary point of the original function subject to the constraints. Unfortunately, the log likelihood function is typically not concave.

Maximum Likelihood

- Given a sample $\{\boldsymbol{x}_n, k_n\}$ of size N from a mixture of multivariate Bernoulli distributions, rewrite it as $\{\boldsymbol{x}_n, \boldsymbol{z}_n\}$ where $\boldsymbol{z}_n$ is a K-dimensional binary vector having only the k_n-th element equal to 1.
- The log likelihood function is

$$\log p(\{\boldsymbol{x}_n, \boldsymbol{z}_n\} | \boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_n \log p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_n \log \prod_k [\pi_k \prod_i \mu_{ki}^{x_{ni}} (1-\mu_{ki})^{(1-x_{ni})}]^{z_{nk}}$$

$$= \sum_n \sum_k z_{nk} [\log \pi_k + \sum_i [x_{ni} \log \mu_{ki} + (1 - x_{ni}) \log(1 - \mu_{ki})]]$$

[1] Any stationary point of the Lagrangian function is a stationary point of the original function subject to the constraints. Unfortunately, the log likelihood function is typically not concave.

Maximum Likelihood

- Given a sample $\{\boldsymbol{x}_n, k_n\}$ of size N from a mixture of multivariate Bernoulli distributions, rewrite it as $\{\boldsymbol{x}_n, \boldsymbol{z}_n\}$ where $\boldsymbol{z}_n$ is a K-dimensional binary vector having only the k_n-th element equal to 1.
- The log likelihood function is

$$\log p(\{\boldsymbol{x}_n, \boldsymbol{z}_n\}|\boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_n \log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_n \log \prod_k [\pi_k \prod_i \mu_{ki}^{x_{ni}} (1-\mu_{ki})^{(1-x_{ni})}]^{z_{nk}}$$

$$= \sum_n \sum_k z_{nk} [\log \pi_k + \sum_i [x_{ni} \log \mu_{ki} + (1 - x_{ni}) \log(1 - \mu_{ki})]]$$

- Let $x'_{ni} = 1 - x_{ni}$ and $\mu'_{ki} = 1 - \mu_{ki}$. To maximize the log likelihood function subject to the constraints $\sum_k \pi_k = 1$ and $\mu_{ki} + \mu'_{ki} = 1$, we maximize

$$\sum_n \sum_k z_{nk} [\log \pi_k + \sum_i [x_{ni} \log \mu_{ki} + x'_{ni} \log \mu'_{ki}]] + \lambda(\sum_k \pi_k - 1) + \sum_k \sum_i \lambda_{ki} (\mu_{ki} + \mu'_{ki} - 1)$$

where λ and λ_{ki} are called Lagrange multipliers.[1]

[1] Any stationary point of the Lagrangian function is a stationary point of the original function subject to the constraints. Unfortunately, the log likelihood function is typically not concave.

Maximum Likelihood

- Given a sample $\{\boldsymbol{x}_n, k_n\}$ of size N from a mixture of multivariate Bernoulli distributions, rewrite it as $\{\boldsymbol{x}_n, \boldsymbol{z}_n\}$ where $\boldsymbol{z}_n$ is a K-dimensional binary vector having only the k_n-th element equal to 1.
- The log likelihood function is

$$\log p(\{\boldsymbol{x}_n, \boldsymbol{z}_n\}|\boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_n \log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_n \log \prod_k [\pi_k \prod_i \mu_{ki}^{x_{ni}} (1-\mu_{ki})^{(1-x_{ni})}]^{z_{nk}}$$

$$= \sum_n \sum_k z_{nk} [\log \pi_k + \sum_i [x_{ni} \log \mu_{ki} + (1 - x_{ni}) \log(1 - \mu_{ki})]]$$

- Let $x'_{ni} = 1 - x_{ni}$ and $\mu'_{ki} = 1 - \mu_{ki}$. To maximize the log likelihood function subject to the constraints $\sum_k \pi_k = 1$ and $\mu_{ki} + \mu'_{ki} = 1$, we maximize

$$\sum_n \sum_k z_{nk} [\log \pi_k + \sum_i [x_{ni} \log \mu_{ki} + x'_{ni} \log \mu'_{ki}]] + \lambda(\sum_k \pi_k - 1) + \sum_k \sum_i \lambda_{ki} (\mu_{ki} + \mu'_{ki} - 1)$$

  where λ and λ_{ki} are called Lagrange multipliers.[1]
- Setting to zero the derivatives with respect to π_k, μ_{ki} and μ'_{ki} gives

$$\pi_k = -\sum_n z_{nk}/\lambda \text{ and } \mu_{ki} = -\sum_n z_{nk} x_{ni}/\lambda_{ki} \text{ and } \mu'_{ki} = -\sum_n z_{nk} x'_{ni}/\lambda_{ki}$$

[1] Any stationary point of the Lagrangian function is a stationary point of the original function subject to the constraints. Unfortunately, the log likelihood function is typically not concave.

Maximum Likelihood

- Given a sample $\{\boldsymbol{x}_n, k_n\}$ of size N from a mixture of multivariate Bernoulli distributions, rewrite it as $\{\boldsymbol{x}_n, \boldsymbol{z}_n\}$ where $\boldsymbol{z}_n$ is a K-dimensional binary vector having only the k_n-th element equal to 1.
- The log likelihood function is

$$\log p(\{\boldsymbol{x}_n, \boldsymbol{z}_n\}|\boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_n \log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_n \log \prod_k [\pi_k \prod_i \mu_{ki}^{x_{ni}} (1-\mu_{ki})^{(1-x_{ni})}]^{z_{nk}}$$

$$= \sum_n \sum_k z_{nk} [\log \pi_k + \sum_i [x_{ni} \log \mu_{ki} + (1 - x_{ni}) \log(1 - \mu_{ki})]]$$

- Let $x'_{ni} = 1 - x_{ni}$ and $\mu'_{ki} = 1 - \mu_{ki}$. To maximize the log likelihood function subject to the constraints $\sum_k \pi_k = 1$ and $\mu_{ki} + \mu'_{ki} = 1$, we maximize

$$\sum_n \sum_k z_{nk} [\log \pi_k + \sum_i [x_{ni} \log \mu_{ki} + x'_{ni} \log \mu'_{ki}]] + \lambda(\sum_k \pi_k - 1) + \sum_k \sum_i \lambda_{ki} (\mu_{ki} + \mu'_{ki} - 1)$$

  where λ and λ_{ki} are called Lagrange multipliers.[1]
- Setting to zero the derivatives with respect to π_k, μ_{ki} and μ'_{ki} gives

$$\pi_k = -\sum_n z_{nk}/\lambda \text{ and } \mu_{ki} = -\sum_n z_{nk} x_{ni}/\lambda_{ki} \text{ and } \mu'_{ki} = -\sum_n z_{nk} x'_{ni}/\lambda_{ki}$$

- Replacing this into the constraint gives $\lambda = -N$ and $\lambda_{ki} = -\sum_n z_{nk}$ and, thus,

$$\pi_k^{ML} = \frac{\sum_n z_{nk}}{N} \text{ and } \mu_{ki}^{ML} = \frac{\sum_n z_{nk} x_{ni}}{\sum_n z_{nk}}$$

[1] Any stationary point of the Lagrangian function is a stationary point of the original function subject to the constraints. Unfortunately, the log likelihood function is typically not concave.

Maximum Likelihood

- Given a sample $\{\boldsymbol{x}_n\}$ of size N from a mixture of multivariate Bernoulli distributions, the expected log likelihood function is

$$\mathbb{E}_{\boldsymbol{Z}}[\log p(\{\boldsymbol{x}_n, \boldsymbol{z}_n\}|\boldsymbol{\mu}, \boldsymbol{\pi})] = \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi}) \log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\mu}, \boldsymbol{\pi})$$

Maximum Likelihood

- Given a sample $\{\boldsymbol{x}_n\}$ of size N from a mixture of multivariate Bernoulli distributions, the expected log likelihood function is

$$\mathbb{E}_{\boldsymbol{Z}}[\log p(\{\boldsymbol{x}_n, \boldsymbol{z}_n\}|\boldsymbol{\mu}, \boldsymbol{\pi})] = \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi}) \log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\mu}, \boldsymbol{\pi})$$

$$= \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi}) \sum_k z_{nk} [\log \pi_k + \sum_i [x_{ni} \log \mu_{ki} + (1 - x_{ni}) \log(1 - \mu_{ki})]]$$

$$= \sum_n \sum_k p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi}) [\log \pi_k + \sum_i [x_{ni} \log \mu_{ki} + (1 - x_{ni}) \log(1 - \mu_{ki})]]$$

Maximum Likelihood

- Given a sample $\{\boldsymbol{x}_n\}$ of size N from a mixture of multivariate Bernoulli distributions, the expected log likelihood function is

$$\mathbb{E}_{\boldsymbol{Z}}[\log p(\{\boldsymbol{x}_n, \boldsymbol{z}_n\}|\boldsymbol{\mu}, \boldsymbol{\pi})] = \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi}) \log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\mu}, \boldsymbol{\pi})$$

$$= \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi}) \sum_k z_{nk}[\log \pi_k + \sum_i [x_{ni} \log \mu_{ki} + (1 - x_{ni}) \log(1 - \mu_{ki})]]$$

$$= \sum_n \sum_k p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})[\log \pi_k + \sum_i [x_{ni} \log \mu_{ki} + (1 - x_{ni}) \log(1 - \mu_{ki})]]$$

- Following a reasoning analogous to the complete-data case, we obtain that

$$\begin{aligned} \pi_k^{ML} &= \frac{\sum_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})}{N} \\ \mu_{ki}^{ML} &= \frac{\sum_n x_{ni} p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})}{\sum_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})} \end{aligned}$$

Maximum Likelihood

- Given a sample $\{\boldsymbol{x}_n\}$ of size N from a mixture of multivariate Bernoulli distributions, the expected log likelihood function is

$$\mathbb{E}_{\boldsymbol{Z}}[\log p(\{\boldsymbol{x}_n, \boldsymbol{z}_n\}|\boldsymbol{\mu}, \boldsymbol{\pi})] = \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi}) \log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\mu}, \boldsymbol{\pi})$$

$$= \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi}) \sum_k z_{nk}[\log \pi_k + \sum_i [x_{ni} \log \mu_{ki} + (1 - x_{ni}) \log(1 - \mu_{ki})]]$$

$$= \sum_n \sum_k p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})[\log \pi_k + \sum_i [x_{ni} \log \mu_{ki} + (1 - x_{ni}) \log(1 - \mu_{ki})]]$$

- Following a reasoning analogous to the complete-data case, we obtain that

$$\pi_k^{ML} = \frac{\sum_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})}{N}$$

$$\mu_{ki}^{ML} = \frac{\sum_n x_{ni} p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})}{\sum_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})}$$

- This is not a closed form solution because

$$p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi}) = \frac{p(z_{nk}, \boldsymbol{x}_n|\boldsymbol{\mu}, \boldsymbol{\pi})}{\sum_k p(z_{nk}, \boldsymbol{x}_n|\boldsymbol{\mu}, \boldsymbol{\pi})} = \frac{\pi_k p(\boldsymbol{x}_n|\boldsymbol{\mu}_k)}{\sum_k \pi_k p(\boldsymbol{x}_n|\boldsymbol{\mu}_k)}$$

but it suggests the following algorithm.

Expectation Maximization Algorithm

EM algorithm

Set $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$ to some initial values
Repeat until $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$ do not change
    Compute $p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})$ for all k and n /* E step */
    Set π_k to π_k^{ML}, and μ_{ki} to μ_{ki}^{ML} for all k and i /* M step */

Expectation Maximization Algorithm

EM algorithm

Set $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$ to some initial values
Repeat until $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$ do not change
    Compute $p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})$ for all k and n /* E step */
    Set π_k to π_k^{ML}, and μ_{ki} to μ_{ki}^{ML} for all k and i /* M step */

- Note that $p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})$ is computed for all k and n in each iteration:

$$p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi}) = \frac{p(z_{nk}, \boldsymbol{x}_n|\boldsymbol{\mu}, \boldsymbol{\pi})}{\sum_k p(z_{nk}, \boldsymbol{x}_n|\boldsymbol{\mu}, \boldsymbol{\pi})} = \frac{\pi_k p(\boldsymbol{x}_n|\boldsymbol{\mu}_k)}{\sum_k \pi_k p(\boldsymbol{x}_n|\boldsymbol{\mu}_k)}$$

Expectation Maximization Algorithm

EM algorithm

Set $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$ to some initial values
Repeat until $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$ do not change
    Compute $p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})$ for all k and n /* E step */
    Set π_k to π_k^{ML}, and μ_{ki} to μ_{ki}^{ML} for all k and i /* M step */

- Note that $p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})$ is computed for all k and n in each iteration:

$$p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi}) = \frac{p(z_{nk}, \boldsymbol{x}_n|\boldsymbol{\mu}, \boldsymbol{\pi})}{\sum_k p(z_{nk}, \boldsymbol{x}_n|\boldsymbol{\mu}, \boldsymbol{\pi})} = \frac{\pi_k p(\boldsymbol{x}_n|\boldsymbol{\mu}_k)}{\sum_k \pi_k p(\boldsymbol{x}_n|\boldsymbol{\mu}_k)}$$

- The difficulty of maximizing the expected log likelihood function is not only that no closed form solution exists, but also that the landscape has typically many local optima. As a result, the EM algorithm is very sensitive to initialization.

Expectation Maximization Algorithm

EM algorithm

Set $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$ to some initial values
Repeat until $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$ do not change
    Compute $p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})$ for all k and n /* E step */
    Set π_k to π_k^{ML}, and μ_{ki} to μ_{ki}^{ML} for all k and i /* M step */

- Note that $p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})$ is computed for all k and n in each iteration:

$$p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi}) = \frac{p(z_{nk}, \boldsymbol{x}_n|\boldsymbol{\mu}, \boldsymbol{\pi})}{\sum_k p(z_{nk}, \boldsymbol{x}_n|\boldsymbol{\mu}, \boldsymbol{\pi})} = \frac{\pi_k p(\boldsymbol{x}_n|\boldsymbol{\mu}_k)}{\sum_k \pi_k p(\boldsymbol{x}_n|\boldsymbol{\mu}_k)}$$

- The difficulty of maximizing the expected log likelihood function is not only that no closed form solution exists, but also that the landscape has typically many local optima. As a result, the EM algorithm is very sensitive to initialization.
- The EM algorithm can also be obtained by maximizing $\log p(\{\boldsymbol{x}_n\}|\boldsymbol{\mu}, \boldsymbol{\pi})$.

Expectation Maximization Algorithm

EM algorithm

Set $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$ to some initial values
Repeat until $\boldsymbol{\pi}$ and $\boldsymbol{\mu}$ do not change
    Compute $p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})$ for all k and n /* E step */
    Set π_k to π_k^{ML}, and μ_{ki} to μ_{ki}^{ML} for all k and i /* M step */

- Note that $p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})$ is computed for all k and n in each iteration:

$$p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi}) = \frac{p(z_{nk}, \boldsymbol{x}_n|\boldsymbol{\mu}, \boldsymbol{\pi})}{\sum_k p(z_{nk}, \boldsymbol{x}_n|\boldsymbol{\mu}, \boldsymbol{\pi})} = \frac{\pi_k p(\boldsymbol{x}_n|\boldsymbol{\mu}_k)}{\sum_k \pi_k p(\boldsymbol{x}_n|\boldsymbol{\mu}_k)}$$

- The difficulty of maximizing the expected log likelihood function is not only that no closed form solution exists, but also that the landscape has typically many local optima. As a result, the EM algorithm is very sensitive to initialization.
- The EM algorithm can also be obtained by maximizing $\log p(\{\boldsymbol{x}_n\}|\boldsymbol{\mu}, \boldsymbol{\pi})$.
- The EM algorithm is guaranteed to increase $\log p(\{\boldsymbol{x}_n\}|\boldsymbol{\mu}, \boldsymbol{\pi})$ in each iteration until a local maximum is reached. So, the algorithm aims for the ML estimates.

Expectation Maximization Algorithm

- We can derive the EM algorithm for mixtures of multivariate Gaussian distributions in much the same way. Simply,

$$
\begin{aligned}
\pi_k^{ML} &= \frac{\sum_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})}{N} \\
\boldsymbol{\mu}_k^{ML} &= \frac{\sum_n \boldsymbol{x}_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})}{\sum_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})} \\
\boldsymbol{\Sigma}_k^{ML} &= \frac{\sum_n (\boldsymbol{x}_n - \boldsymbol{\mu}_k^{ML})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{ML})^T p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})}{\sum_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})}
\end{aligned}
$$

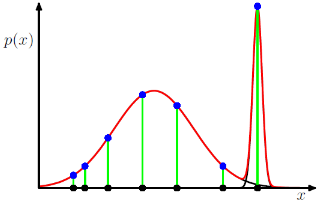
Expectation Maximization Algorithm

- We can derive the EM algorithm for mixtures of multivariate Gaussian distributions in much the same way. Simply,

$$\begin{aligned}
\pi_k^{ML} &= \frac{\sum_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})}{N} \\
\boldsymbol{\mu}_k^{ML} &= \frac{\sum_n \boldsymbol{x}_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})}{\sum_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})} \\
\boldsymbol{\Sigma}_k^{ML} &= \frac{\sum_n (\boldsymbol{x}_n - \boldsymbol{\mu}_k^{ML})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{ML})^T p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})}{\sum_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})}
\end{aligned}$$

- Unlike in the case of mixture of multivariate Bernoulli distributions, there can be singularities, i.e. the log likelihood function goes to infinity when a component of the mixture collapses onto a single data point.

Figure 9.7 Illustration of how singularities in the likelihood function arise with mixtures of Gaussians. This should be compared with the case of a single Gaussian shown in Figure 1.14 for which no singularities arise.



Expectation Maximization Algorithm

- We can derive the EM algorithm for mixtures of multivariate Gaussian distributions in much the same way. Simply,

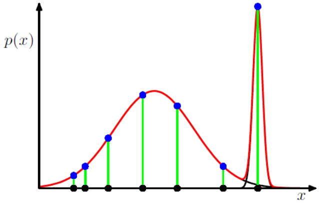
$$\pi_k^{ML} = \frac{\sum_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})}{N}$$

$$\boldsymbol{\mu}_k^{ML} = \frac{\sum_n \boldsymbol{x}_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})}{\sum_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})}$$

$$\boldsymbol{\Sigma}_k^{ML} = \frac{\sum_n (\boldsymbol{x}_n - \boldsymbol{\mu}_k^{ML})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{ML})^T p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})}{\sum_n p(z_{nk}|\boldsymbol{x}_n, \boldsymbol{\mu}, \boldsymbol{\pi})}$$

- Unlike in the case of mixture of multivariate Bernoulli distributions, there can be singularities, i.e. the log likelihood function goes to infinity when a component of the mixture collapses onto a single data point.

Figure 9.7 Illustration of how singularities in the likelihood function arise with mixtures of Gaussians. This should be compared with the case of a single Gaussian shown in Figure 1.14 for which no singularities arise.



- Solution: Reset the mean and covariance of the component to random and large values, respectively. Or adopt a Bayesian approach.

Number of Mixture Components

- Too few/many components will result in underfitting/overfitting.

Number of Mixture Components

- Too few/many components will result in underfitting/overfitting.
- We can perform a search over the number of components by scoring each number with, for instance, the Bayesian information criterion (BIC):

$$\log p(\{\boldsymbol{x}_n\}|\boldsymbol{\mu}^{ML}, \boldsymbol{\pi}^{ML}) - \frac{M}{2} \log N$$

  where M is the number of free parameters in the mixture model. Note that the EM algorithm has to be run for each candidate number.

Number of Mixture Components

- Too few/many components will result in underfitting/overfitting.
- We can perform a search over the number of components by scoring each number with, for instance, the Bayesian information criterion (BIC):

$$\log p(\{\boldsymbol{x}_n\}|\boldsymbol{\mu}^{ML}, \boldsymbol{\pi}^{ML}) - \frac{M}{2} \log N$$

  where M is the number of free parameters in the mixture model. Note that the EM algorithm has to be run for each candidate number.
- Under some conditions, the score above can be seen as an approximation of the Bayesian score for a given number of components:

$$\log p(\{\boldsymbol{x}_n\}) = \int \int \log p(\{\boldsymbol{x}_n\}|\boldsymbol{\mu}, \boldsymbol{\pi}) p(\boldsymbol{\mu}, \boldsymbol{\pi}) d\boldsymbol{\mu} d\boldsymbol{\pi}$$

Number of Mixture Components

- Too few/many components will result in underfitting/overfitting.
- We can perform a search over the number of components by scoring each number with, for instance, the Bayesian information criterion (BIC):

$$\log p(\{\boldsymbol{x}_n\}|\boldsymbol{\mu}^{ML}, \boldsymbol{\pi}^{ML}) - \frac{M}{2} \log N$$

  where M is the number of free parameters in the mixture model. Note that the EM algorithm has to be run for each candidate number.
- Under some conditions, the score above can be seen as an approximation of the Bayesian score for a given number of components:

$$\log p(\{\boldsymbol{x}_n\}) = \int \int \log p(\{\boldsymbol{x}_n\}|\boldsymbol{\mu}, \boldsymbol{\pi}) p(\boldsymbol{\mu}, \boldsymbol{\pi}) d\boldsymbol{\mu} d\boldsymbol{\pi}$$

- There also exist algorithms that iteratively split and merge components according to the scores above until no further improvement occurs.

Number of Mixture Components

- Too few/many components will result in underfitting/overfitting.
- We can perform a search over the number of components by scoring each number with, for instance, the Bayesian information criterion (BIC):

$$\log p(\{\boldsymbol{x}_n\}|\boldsymbol{\mu}^{ML}, \boldsymbol{\pi}^{ML}) - \frac{M}{2} \log N$$

  where M is the number of free parameters in the mixture model. Note that the EM algorithm has to be run for each candidate number.
- Under some conditions, the score above can be seen as an approximation of the Bayesian score for a given number of components:

$$\log p(\{\boldsymbol{x}_n\}) = \int \int \log p(\{\boldsymbol{x}_n\}|\boldsymbol{\mu}, \boldsymbol{\pi}) p(\boldsymbol{\mu}, \boldsymbol{\pi}) d\boldsymbol{\mu} d\boldsymbol{\pi}$$

- There also exist algorithms that iteratively split and merge components according to the scores above until no further improvement occurs.
- Nested cross-validations is also an option.

Model-Based Clustering

- A mixture model represents a mixture of subpopulations, a.k.a. clusters:
  1. Choose a subpopulation according to $\mathit{Multinomial}(k|\pi_1, \ldots, \pi_K)$.
  2. Sample an instance from the chosen subpopulation according to $p(\boldsymbol{x}|\boldsymbol{\mu}_k)$.

Model-Based Clustering

- A mixture model represents a mixture of subpopulations, a.k.a. clusters:
  1. Choose a subpopulation according to $\mathit{Multinomial}(k|\pi_1, \ldots, \pi_K)$.
  2. Sample an instance from the chosen subpopulation according to $p(\boldsymbol{x}|\boldsymbol{\mu}_k)$.
- Model-based clustering aims to soft-assign instances to the different subpopulations by computing

$$p(k|\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\pi}) = \frac{\pi_k p(\boldsymbol{x}|\boldsymbol{\mu}_k)}{\sum_k \pi_k p(\boldsymbol{x}|\boldsymbol{\mu}_k)}$$

Model-Based Clustering

- A mixture model represents a mixture of subpopulations, a.k.a. clusters:
  1. Choose a subpopulation according to $\mathit{Multinomial}(k|\pi_1, \ldots, \pi_K)$.
  2. Sample an instance from the chosen subpopulation according to $p(\boldsymbol{x}|\boldsymbol{\mu}_k)$.
- Model-based clustering aims to soft-assign instances to the different subpopulations by computing

$$p(k|\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\pi}) = \frac{\pi_k p(\boldsymbol{x}|\boldsymbol{\mu}_k)}{\sum_k \pi_k p(\boldsymbol{x}|\boldsymbol{\mu}_k)}$$

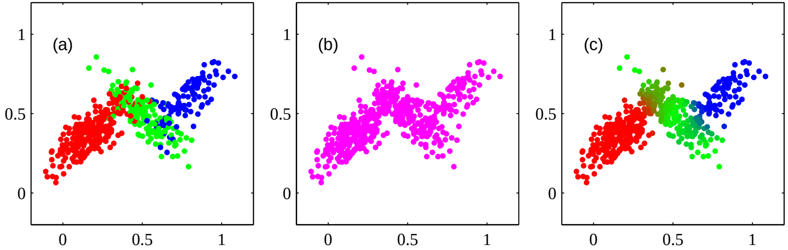
- To do so, the number of subpopulations and their parameters must be estimated: The EM algorithm and BIC score.

Model-Based Clustering

- A mixture model represents a mixture of subpopulations, a.k.a. clusters:
  1. Choose a subpopulation according to $\mathit{Multinomial}(k|\pi_1, \dots, \pi_K)$.
  2. Sample an instance from the chosen subpopulation according to $p(\mathbf{x}|\boldsymbol{\mu}_k)$.
- Model-based clustering aims to soft-assign instances to the different subpopulations by computing

$$p(k|\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\pi}) = \frac{\pi_k p(\mathbf{x}|\boldsymbol{\mu}_k)}{\sum_k \pi_k p(\mathbf{x}|\boldsymbol{\mu}_k)}$$

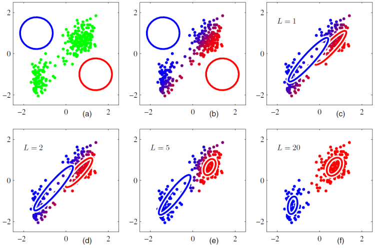
- To do so, the number of subpopulations and their parameters must be estimated: The EM algorithm and BIC score.



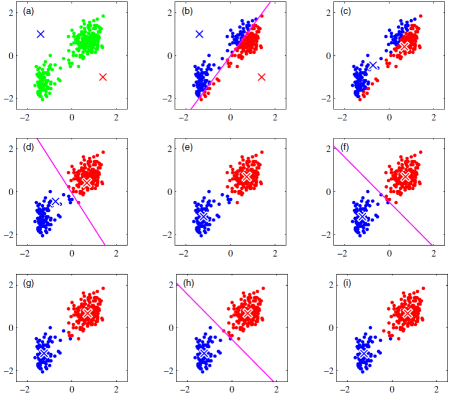
(a) Sample with cluster labels, (b) initial clustering, and (c) final clustering.

K-Means Algorithm

```
Assign each point to a cluster (a.k.a subpopulation) at random
Compute the cluster centroids as the averages of the points assigned to each cluster
Repeat until the centroids do not change
    Assign each point to the cluster with the closest centroid
    Update the cluster centroids as the averages of the points assigned to each cluster
```



EM algorithm



K-means algorithm

K-Means Algorithm

- Recall that

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{2\pi^{D/2}} \frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}} \exp(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k))$$

K-Means Algorithm

- Recall that

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{2\pi^{D/2}} \frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}} \exp(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k))$$

- Assume that $\boldsymbol{\Sigma}_k = \epsilon \boldsymbol{I}$ where ϵ is a variance parameter and $\boldsymbol{I}$ is the identity matrix. Then,

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{2\pi^{D/2}} \frac{1}{|\epsilon \boldsymbol{I}|^{1/2}} \exp(-\frac{1}{2\epsilon} ||\boldsymbol{x} - \boldsymbol{\mu}_k||^2)$$

K-Means Algorithm

- Recall that

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{2\pi^{D/2}} \frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}} \exp(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k))$$

- Assume that $\boldsymbol{\Sigma}_k = \epsilon \boldsymbol{I}$ where ϵ is a variance parameter and $\boldsymbol{I}$ is the identity matrix. Then,

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{2\pi^{D/2}} \frac{1}{|\epsilon \boldsymbol{I}|^{1/2}} \exp(-\frac{1}{2\epsilon} ||\boldsymbol{x} - \boldsymbol{\mu}_k||^2)$$

$$p(k|\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\pi}) = \frac{\pi_k p(\boldsymbol{x}|\boldsymbol{\mu}_k)}{\sum_k \pi_k p(\boldsymbol{x}|\boldsymbol{\mu}_k)} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_k \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} = \frac{\pi_k \exp(-\frac{1}{2\epsilon} ||\boldsymbol{x} - \boldsymbol{\mu}_k||^2)}{\sum_k \pi_k \exp(-\frac{1}{2\epsilon} ||\boldsymbol{x} - \boldsymbol{\mu}_k||^2)}$$

K-Means Algorithm

- Recall that

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{2\pi^{D/2}} \frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}} \exp(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k))$$

- Assume that $\boldsymbol{\Sigma}_k = \epsilon \boldsymbol{I}$ where ϵ is a variance parameter and $\boldsymbol{I}$ is the identity matrix. Then,

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{2\pi^{D/2}} \frac{1}{|\epsilon \boldsymbol{I}|^{1/2}} \exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_k||^2)$$

$$p(k|\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\pi}) = \frac{\pi_k p(\boldsymbol{x}|\boldsymbol{\mu}_k)}{\sum_k \pi_k p(\boldsymbol{x}|\boldsymbol{\mu}_k)} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_k \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} = \frac{\pi_k \exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_k||^2)}{\sum_k \pi_k \exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_k||^2)}$$

- As $\epsilon \to 0$, the smaller $||\boldsymbol{x}-\boldsymbol{\mu}_k||^2$ the slower $\exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_k||^2)$ goes to 0.

K-Means Algorithm

- Recall that

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{2\pi^{D/2}} \frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}} \exp(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k))$$

- Assume that $\boldsymbol{\Sigma}_k = \epsilon \boldsymbol{I}$ where ϵ is a variance parameter and $\boldsymbol{I}$ is the identity matrix. Then,

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{2\pi^{D/2}} \frac{1}{|\epsilon \boldsymbol{I}|^{1/2}} \exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_k||^2)$$

$$p(k|\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\pi}) = \frac{\pi_k p(\boldsymbol{x}|\boldsymbol{\mu}_k)}{\sum_k \pi_k p(\boldsymbol{x}|\boldsymbol{\mu}_k)} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_k \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} = \frac{\pi_k \exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_k||^2)}{\sum_k \pi_k \exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_k||^2)}$$

- As $\epsilon \to 0$, the smaller $||\boldsymbol{x}-\boldsymbol{\mu}_k||^2$ the slower $\exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_k||^2)$ goes to 0.
- As $\epsilon \to 0$, instances are hard-assigned (i.e. with probability 1) to the subpopulation with closest mean. This clustering technique is known as K-means algorithm.

K-Means Algorithm

- Recall that

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{2\pi^{D/2}} \frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}} \exp(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k))$$

- Assume that $\boldsymbol{\Sigma}_k = \epsilon \boldsymbol{I}$ where ϵ is a variance parameter and $\boldsymbol{I}$ is the identity matrix. Then,

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{2\pi^{D/2}} \frac{1}{|\epsilon \boldsymbol{I}|^{1/2}} \exp(-\frac{1}{2\epsilon} ||\boldsymbol{x} - \boldsymbol{\mu}_k||^2)$$

$$p(k|\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\pi}) = \frac{\pi_k p(\boldsymbol{x}|\boldsymbol{\mu}_k)}{\sum_k \pi_k p(\boldsymbol{x}|\boldsymbol{\mu}_k)} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_k \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} = \frac{\pi_k \exp(-\frac{1}{2\epsilon}||\boldsymbol{x} - \boldsymbol{\mu}_k||^2)}{\sum_k \pi_k \exp(-\frac{1}{2\epsilon}||\boldsymbol{x} - \boldsymbol{\mu}_k||^2)}$$

- As $\epsilon \to 0$, the smaller $||\boldsymbol{x} - \boldsymbol{\mu}_k||^2$ the slower $\exp(-\frac{1}{2\epsilon}||\boldsymbol{x} - \boldsymbol{\mu}_k||^2)$ goes to 0.
- As $\epsilon \to 0$, instances are hard-assigned (i.e. with probability 1) to the subpopulation with closest mean. This clustering technique is known as K-means algorithm.
- Note that $\boldsymbol{\pi}$ and $\boldsymbol{\Sigma}_k$ play no role in the K-means algorithm whereas, in each iteration, $\boldsymbol{\mu}_k$ is updated to the average of the instances assigned to subpopulation k.

K-Means Algorithm

- Recall that

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{2\pi^{D/2}} \frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right)$$

- Assume that $\boldsymbol{\Sigma}_k = \epsilon \boldsymbol{I}$ where ϵ is a variance parameter and $\boldsymbol{I}$ is the identity matrix. Then,

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{2\pi^{D/2}} \frac{1}{|\epsilon \boldsymbol{I}|^{1/2}} \exp\left(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_k||^2\right)$$

$$p(k|\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\pi}) = \frac{\pi_k p(\boldsymbol{x}|\boldsymbol{\mu}_k)}{\sum_k \pi_k p(\boldsymbol{x}|\boldsymbol{\mu}_k)} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_k \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} = \frac{\pi_k \exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_k||^2)}{\sum_k \pi_k \exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_k||^2)}$$

- As $\epsilon \to 0$, the smaller $||\boldsymbol{x}-\boldsymbol{\mu}_k||^2$ the slower $\exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_k||^2)$ goes to 0.
- As $\epsilon \to 0$, instances are hard-assigned (i.e. with probability 1) to the subpopulation with closest mean. This clustering technique is known as K-means algorithm.
- Note that $\boldsymbol{\pi}$ and $\boldsymbol{\Sigma}_k$ play no role in the K-means algorithm whereas, in each iteration, $\boldsymbol{\mu}_k$ is updated to the average of the instances assigned to subpopulation k.
- The K-means algorithm can be used to initialize the EM algorithm.

Summary

- Mixture models: To model complex distributions by linearly combining simple distributions.

Summary

- Mixture models: To model complex distributions by linearly combining simple distributions.
- EM algorithm: To estimate the ML parameters of mixture models. It converges to a local maximum of the log likelihood of the observed data.

Summary

- Mixture models: To model complex distributions by linearly combining simple distributions.
- EM algorithm: To estimate the ML parameters of mixture models. It converges to a local maximum of the log likelihood of the observed data.
- We can see mixture models as model-based clustering, and the K-means algorithm as a limit case thereof.

Summary

- Mixture models: To model complex distributions by linearly combining simple distributions.
- EM algorithm: To estimate the ML parameters of mixture models. It converges to a local maximum of the log likelihood of the observed data.
- We can see mixture models as model-based clustering, and the K-means algorithm as a limit case thereof.
- The EM algorithm can be used to estimate the ML parameters from data with any pattern of missing (at random) entries, i.e. not only one latent variable.