Clustering, Mixture Models, and EM Algorithm

Man-Wai MAK

Dept. of Electronic and Information Engineering,
The Hong Kong Polytechnic University

enmwmak@polyu.edu.hk
http://www.eie.polyu.edu.hk/∼mwmak

References:

- C. Bishop, *Pattern Recognition and Machine Learning*, Chapter 9, Springer, 2006.
- S.Y. Kung, M.W. Mak and S.H. Lin, *Biometric Authentication: A Machine Learning Approach*, Chapter 3, Prentice Hall, 2005.
- M.W. Mak and J.T. Chien, *Machine Learning for Speaker Recognition*, Cambridge University Press, 2020.

October 18, 2019

Overview

Motivations

- Clustering is a kind of unsupervised learning, which has been used in many disciplines.
  - **Power Electronics**: "Genetic k-means algorithm based RBF network for photovoltaic MPP prediction." *Energy*, 35.2 (2010): 529-536.
  - **Telecommunication**: "An energy efficient hierarchical clustering algorithm for wireless sensor networks." *INFOCOM 2003*, Vol. 3. IEEE, 2003.
  - **Photonics**: "Contiguity-enhanced k-means clustering algorithm for unsupervised multispectral image segmentation." *Optical Science, Engineering and Instrumentation'97*, International Society for Optics and Photonics, 1997.
  - **Multimedia**: "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, Aug 2000.

K-means

- Divide a data set $\mathcal{X} = \{\mathbf{x}_t; t = 1, \ldots, T\}$ into K groups, each represented by its centroid denoted by $\boldsymbol{\mu}_k$, $k = 1, \ldots, K$.
- The task is
  1. to determine the K centroids $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$ and
  2. to assign each pattern $\mathbf{x}_t$ to **one** of the centroids.
- Mathematically speaking, one denotes the centroid associated with $\mathbf{x}_t$ as $\mathbf{c}_t$, where $\mathbf{c}_t \in \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$.
- Then the objective of the K-means algorithm is to minimize the sum of squared errors:

$$\begin{aligned} E(\mathcal{X}) &= \sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{c}_t\|^2 \\ &= \sum_{t=1}^{T} (\mathbf{x}_t - \mathbf{c}_t)^{\mathsf{T}} (\mathbf{x}_t - \mathbf{c}_t). \end{aligned} \tag{1}$$

K-means

- Let $\mathcal{X}_k$ denotes the set of data vectors associated with the k-th cluster with the centroid $\boldsymbol{\mu}_k$ and N_k denotes the number of vectors in it.
- The learning rule of the K-means algorithm consists of:
  1. *Determine the membership of a data vector:*

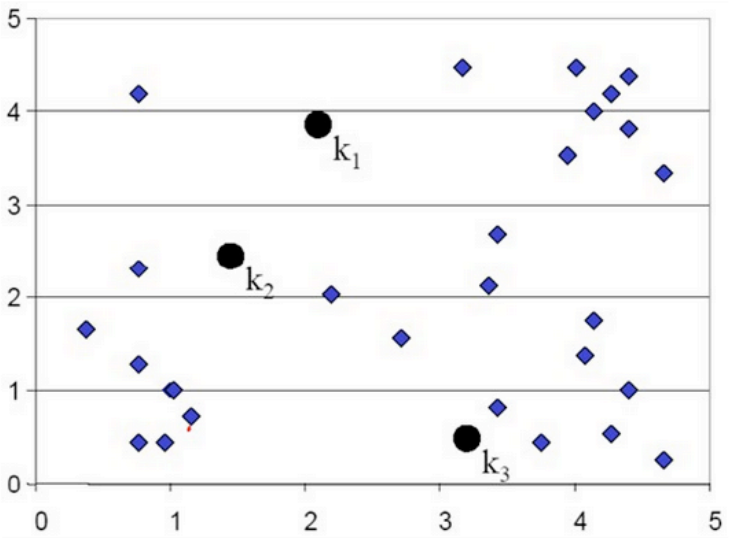
$$\mathbf{x} \in \mathcal{X}_k \quad \text{if} \quad \|\mathbf{x} - \boldsymbol{\mu}_k\| < \|\mathbf{x} - \boldsymbol{\mu}_j\| \quad \forall j \neq k. \tag{2}$$

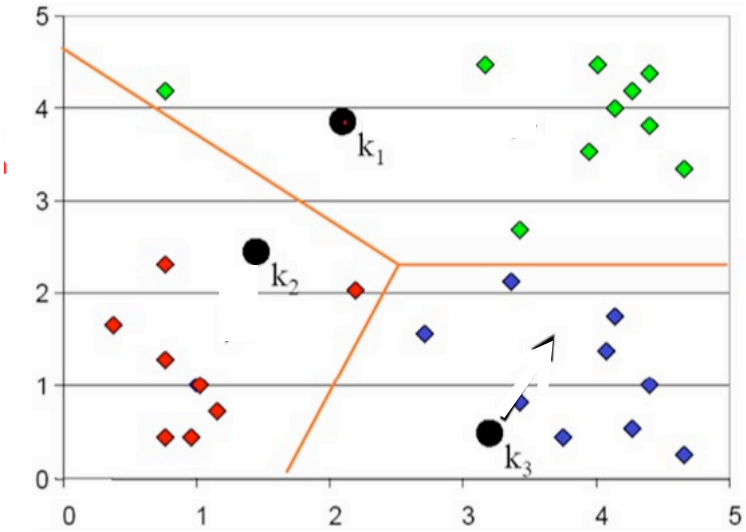
  2. *Update the representation of the cluster:* The centroid is updated based on the new membership:

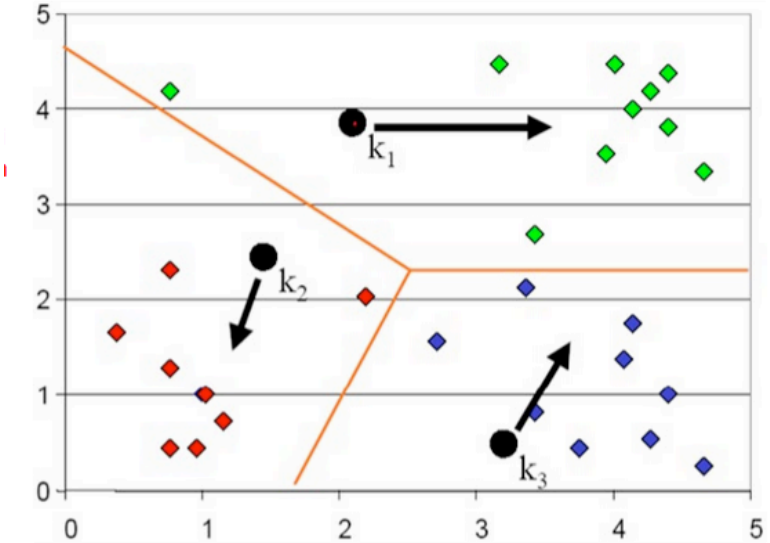
$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{\mathbf{x} \in \mathcal{X}_k} \mathbf{x}, \quad k = 1, \ldots, K. \tag{3}$$

K-means

- K-means procedure:







K-means

- K-means procedure:
  1. Randomly picks K samples from the training data and consider them as the centroids. In the example on previous page, $K = 3$.
  2. For each training sample, assign it to the nearest centroid. In this example, samples are assigned to either green, red or blue diamond.
  3. For each cluster (green, red, or blue), re-compute the cluster means. Then, repeat step 2 until no change in the centroids.

Example Applications of K-means

- Assume that we got some iris flowers

Setosa

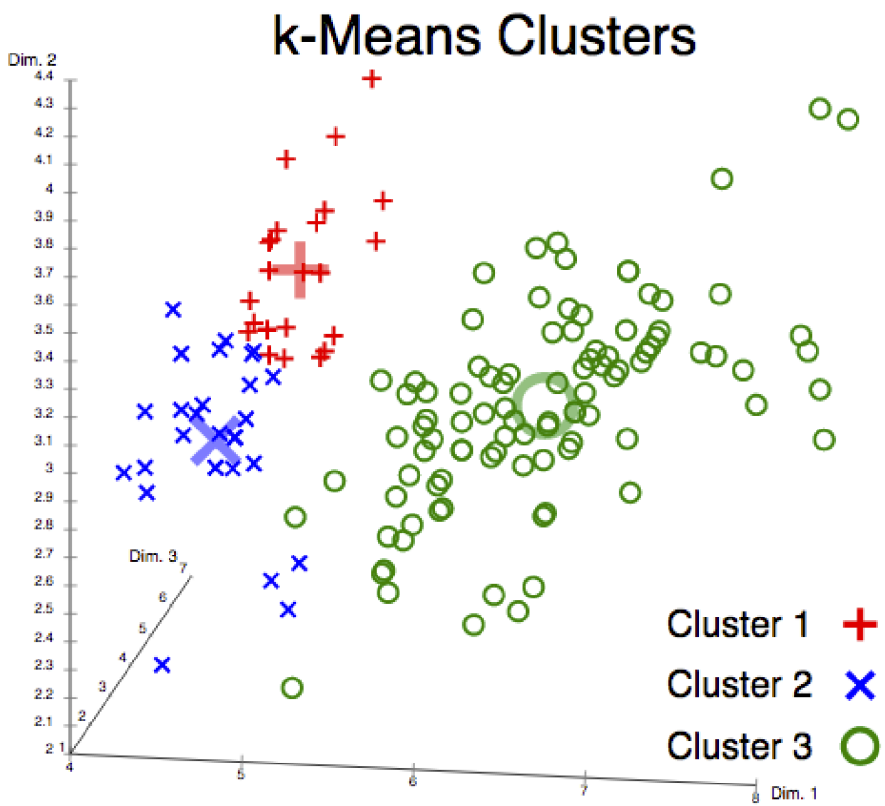
Versicolor

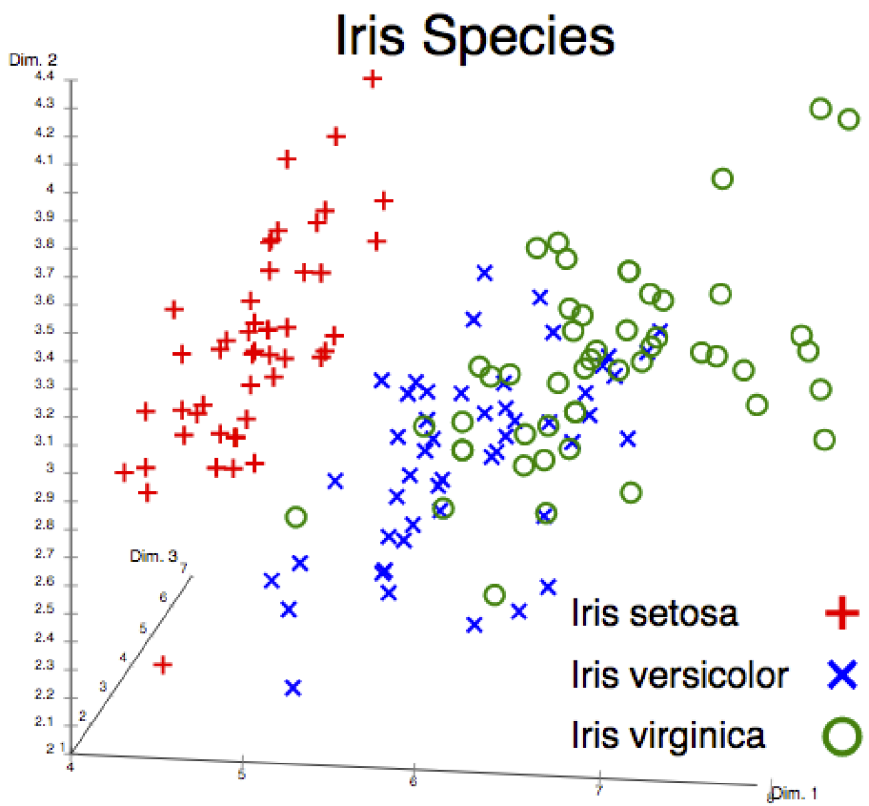
Virginica

- Four attributes (features): (1) sepal length, (2) sepal width, (3) petal length, and (4) petal width
- We only know there are 3 types of iris flowers but no labels are available in the dataset.
- We may apply K-means to divide the 4-dimensional vectors into 3 clusters.
- But we still do not know which cluster belongs to which iris type.

Example Applications of K-means

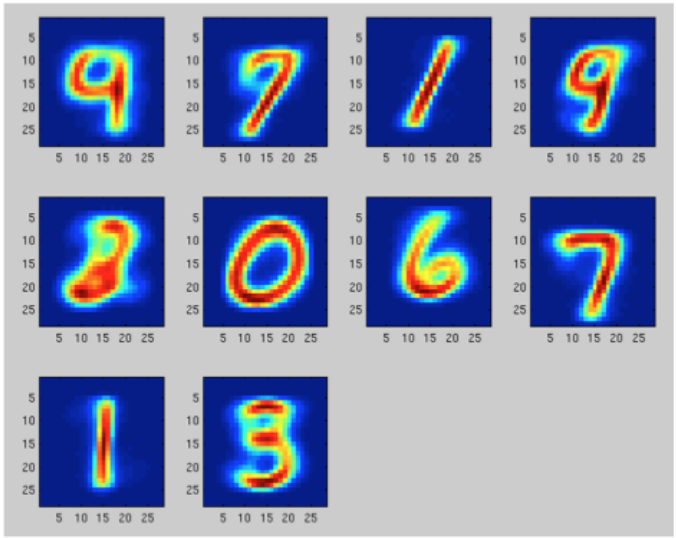
- Results of K-mean clustering:

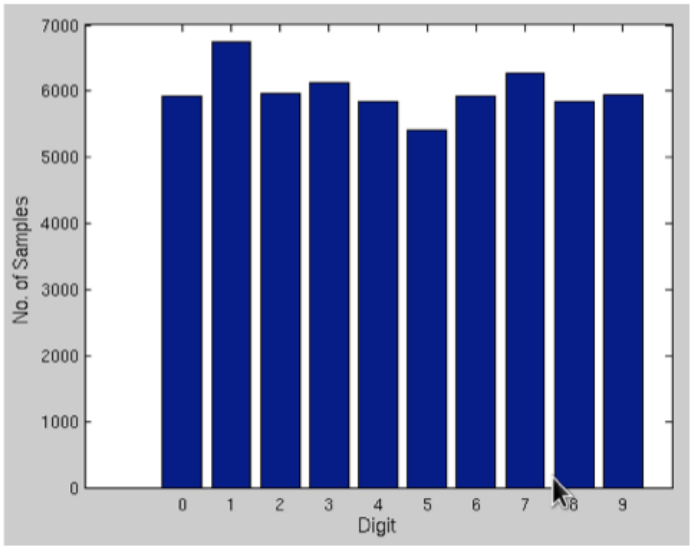




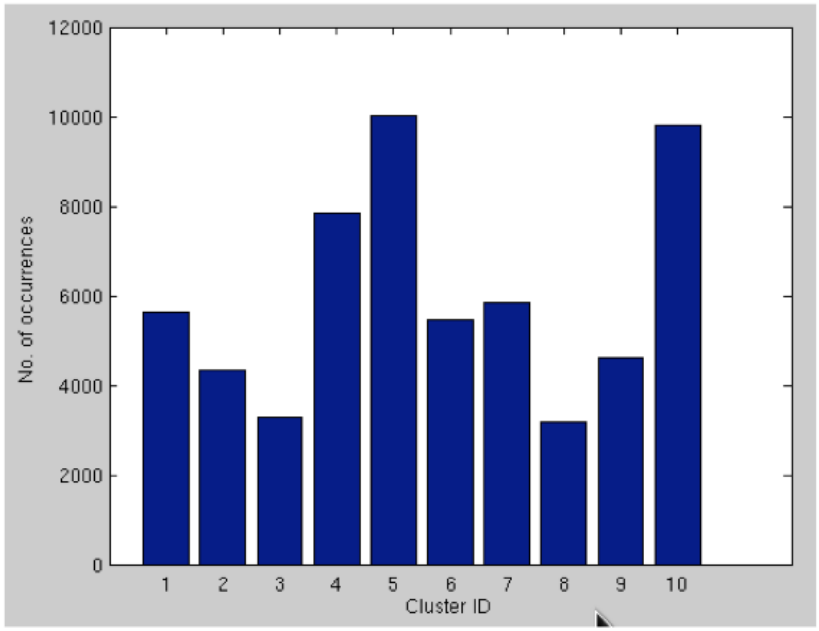
Example Applications of K-means

- K-mean Clustering of handwritten digits with $K = 10$





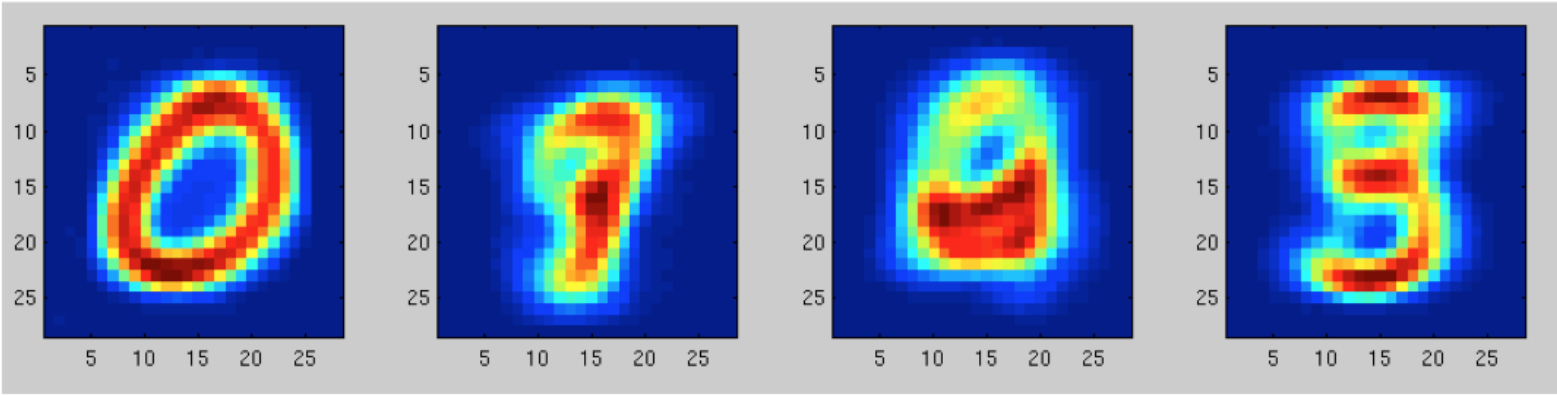
No. of samples for each cluster

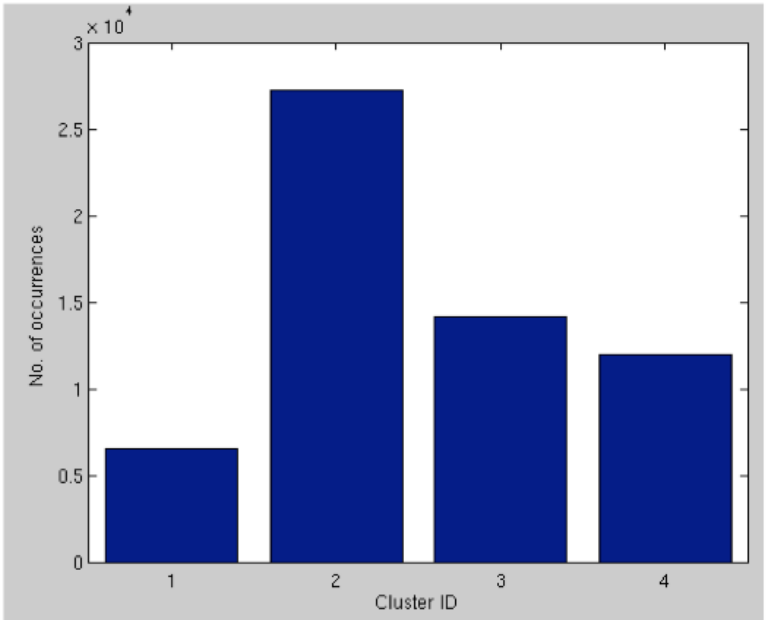


No. of samples for each digit in the training set

Example Applications of K-means

- K-mean Clustering of handwritten digits with $K = 4$





No. of samples for each cluster

Gaussian Mixture Models (GMM)

Gaussian Mixture Models

- A Gaussian mixture model (GMM) is a linear weighted sum of K Gaussian densities:

$$p(\mathbf{x}) = \sum_{k=1}^{K} w_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

where $w_k \equiv Pr(\text{mix} = k)$ is the k-th mixture coefficient and

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

is the k-th Gaussian density with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$.

- Note that $\sum_{k=1}^{K} w_k = 1$.

Gaussian Mixture Models

- GMM with 3 mixtures ($K = 3$):

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

$$p(\mathbf{x}) = \sum_{k=1}^{K} w_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

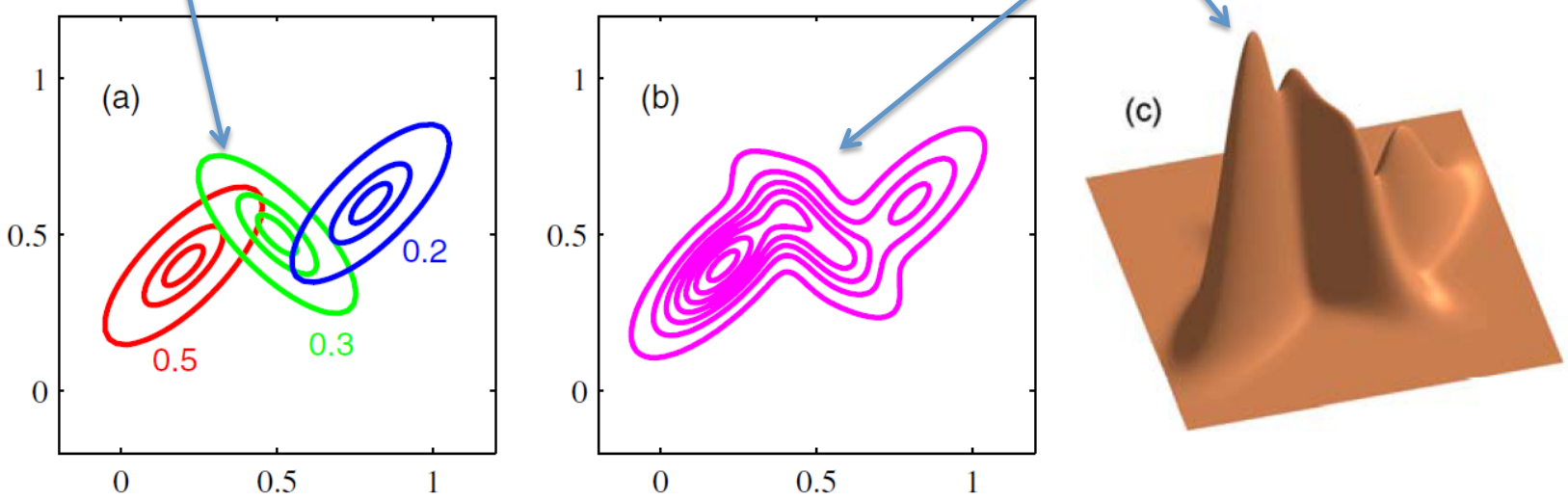
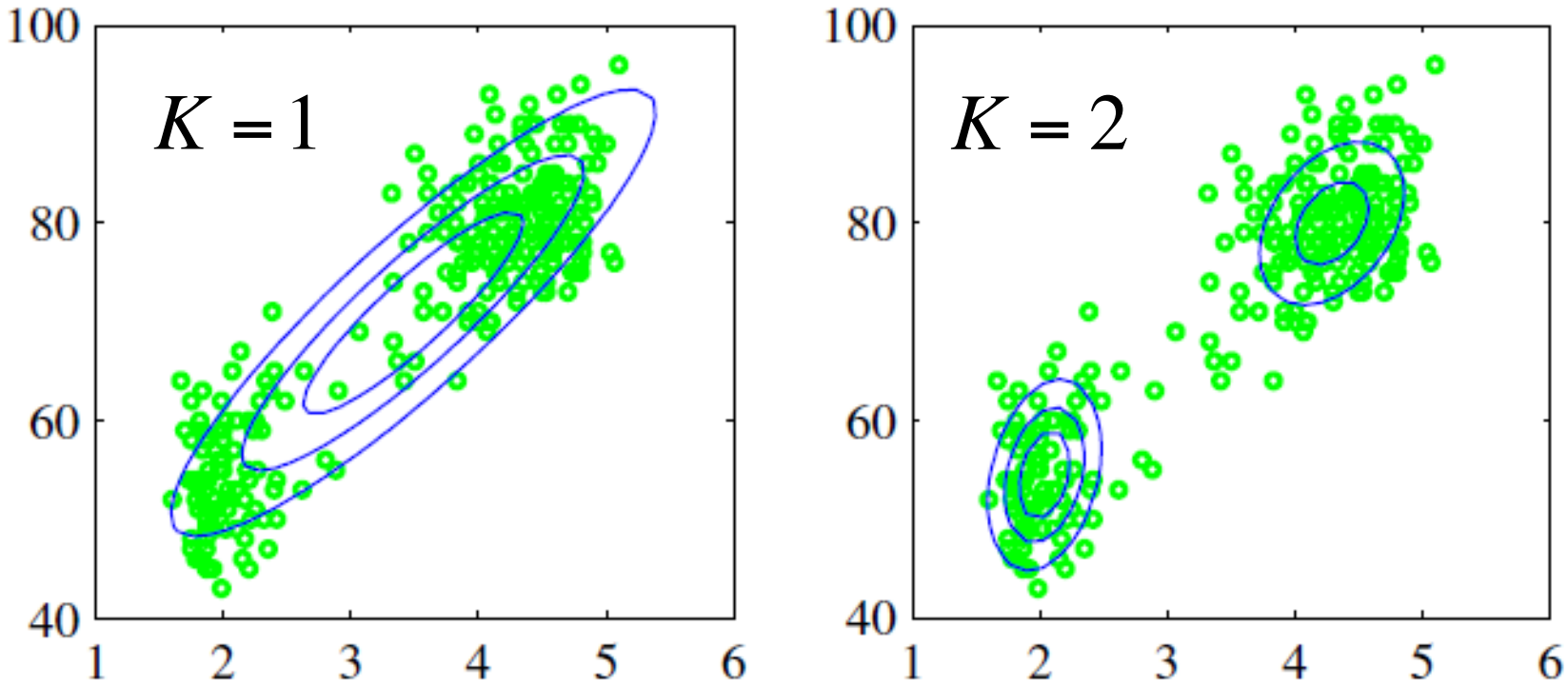


Figure 2.23 Illustration of a mixture of 3 Gaussians in a two-dimensional space. (a) Contours of constant density for each of the mixture components, in which the 3 components are denoted red, blue and green, and the values of the mixing coefficients are shown below each component. (b) Contours of the marginal probability density $p(\mathbf{x})$ of the mixture distribution. (c) A surface plot of the distribution $p(\mathbf{x})$.

Gaussian Mixture Models

- GMM clustering:



Training of GMM by Maximum Likelihood

- Given a set of N-independent and identically distributed (iid) vectors $\mathcal{X} = \{\mathbf{x}_n; n = 1, \ldots, N\}$, the log of the likelihood function is given by

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

- To find the parameters $\theta = \{w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$ that maximize $\log p(\mathcal{X}|\theta)$, we may set $\frac{\partial \log p(\mathcal{X})}{\partial \theta} = 0$ and solve for θ.
- But this method will not give a closed-form solution for θ.
- The trouble is that the summation appears inside the logarithm.

Training of GMM by Maximum Likelihood

- An elegant method for finding maximum-likelihood solutions for model with latent variable is the expectation-maximization (EM) algorithm.
- In GMM, for each data point $\mathbf{x}_n$, we do not know which Gaussian generates it. So, the latent information is the Gaussian ID for each $\mathbf{x}_n$.
- Define $\mathcal{Z} = \{z_{nk}; n = 1, \ldots, N; k = 1, \ldots, K\}$ as the set of latent variables, where $z_{nk} = 1$ if $\mathbf{x}_n$ is generated by the k-th Gaussian; otherwise $z_{nk} = 0$.
- $\{\mathcal{X}, \mathcal{Z}\}$ is called the *complete* data set, and $\mathcal{X}$ is the *incomplete* data set.
- In most cases, including GMM, maximizing $\log p(\mathcal{X}, \mathcal{Z}|\theta)$ with respect to θ is straightforward.
- Fig. 9.5(a) [next page] shows the distribution $p(\mathbf{x}, \mathbf{z})$ of the complete data, whereas Fig. 9.5(b) shows the distribution $p(\mathbf{x})$ of the incomplete data.

GMM Joint vs Marginal Distributions

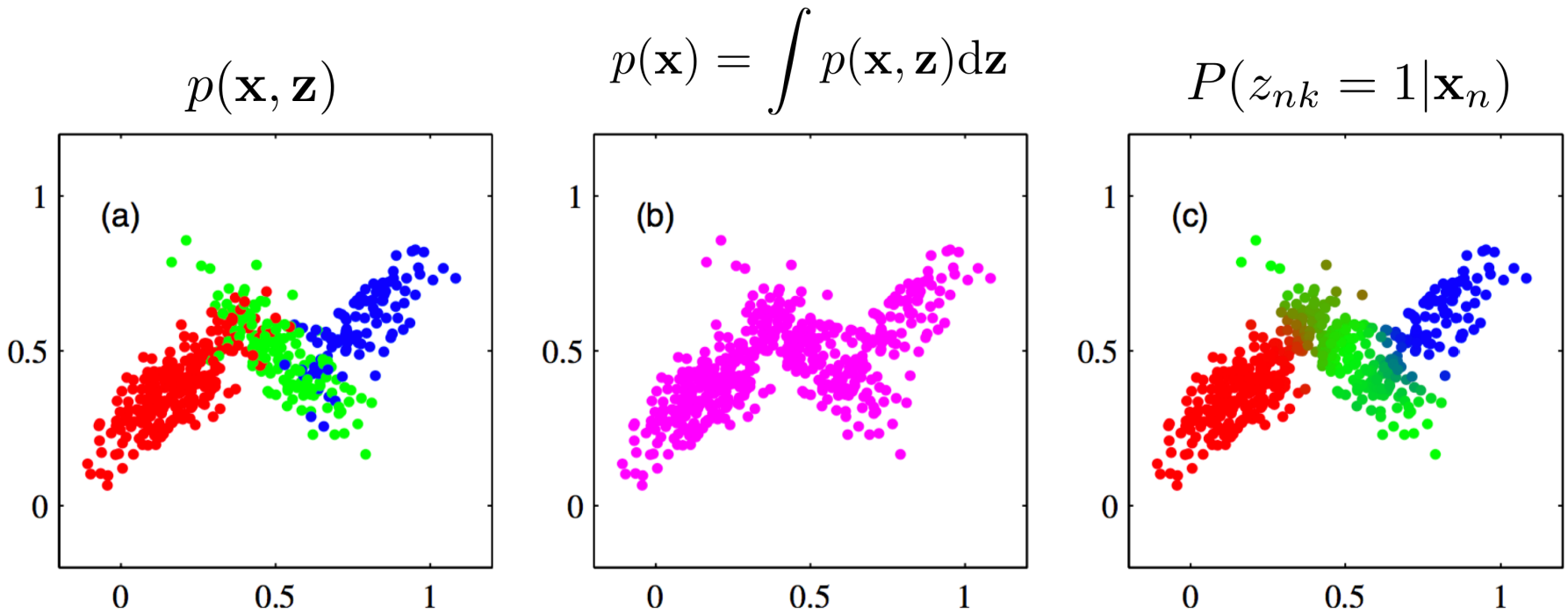


Figure 9.5 Example of 500 points drawn from the mixture of 3 Gaussians shown in Figure 2.23. (a) Samples from the joint distribution $p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$ in which the three states of $\mathbf{z}$, corresponding to the three components of the mixture, are depicted in red, green, and blue, and (b) the corresponding samples from the marginal distribution $p(\mathbf{x})$, which is obtained by simply ignoring the values of $\mathbf{z}$ and just plotting the $\mathbf{x}$ values. The data set in (a) is said to be *complete*, whereas that in (b) is *incomplete*. (c) The same samples in which the colours represent the value of the responsibilities $\gamma(z_{nk})$ associated with data point $\mathbf{x}_n$, obtained by plotting the corresponding point using proportions of red, blue, and green ink given by $\gamma(z_{nk})$ for $k = 1, 2, 3$, respectively

Source: C.M. Bishop (2006)

EM Algorithm for GMM

- However, we actually do not know $\mathcal{Z}$. So, we could not compute $\ln p(\mathcal{Z}, \mathcal{X}|\theta)$.
- Fortunately, we know its posterior distribution, i.e., $P(\mathcal{Z}|\mathcal{X}, \theta)$, through the Bayes theorem:[1]

$$P(z|x) = \frac{P(z)p(x|z)}{p(x)}$$

- In the context of GMM, we compute the posterior probability for each $\mathbf{x}_n$:

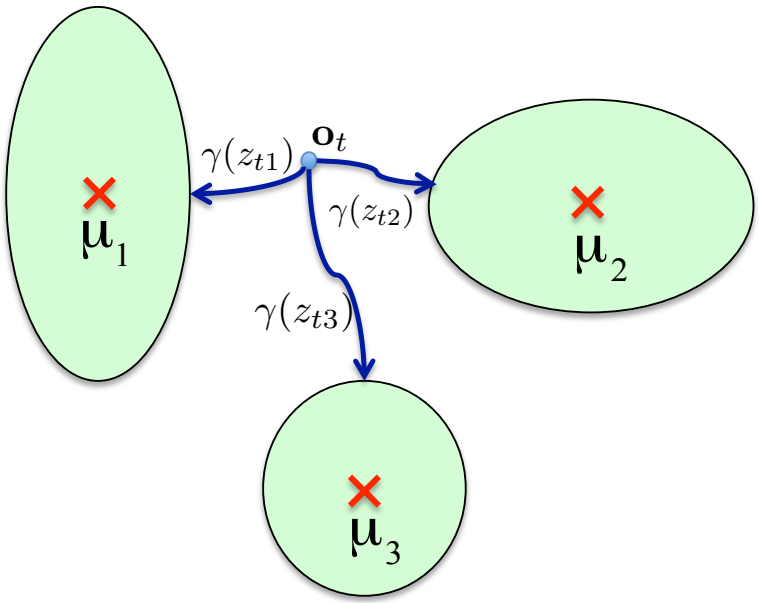
$$\gamma(z_{nk}) \equiv P(z_{nk} = 1|\mathbf{x}_n, \theta) = \frac{w_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} w_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{4}$$

- Eq. 4 constitutes the E-step of the EM algorithm.

[1] We denote probabilities and probability mass functions of discrete random variable using capital letter P.

EM Algorithm for GMM

- Computing the posteriors of the latent variables can be considered as *alignment*.
- The posterior probabilities indicate the *closeness* of $\mathbf{x}_n$ to individual Gaussians in the Mahalanobis sense.



- Mahalanobis distance between $\mathbf{x}$ and $\mathbf{y}$ is

$$D_{\text{mah}}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \mathbf{y})}$$

EM Algorithm for GMM

- So, given the current estimate of the model parameters θ^{old}, we can find its new estimate θ by computing the expected value of $\ln p(\mathcal{Z}, \mathcal{X}|\theta)$ under the posterior distribution of $\mathcal{Z}$:

$$
\begin{aligned}
Q(\theta|\theta^{\text{old}}) &= \mathbb{E}_{\mathcal{Z}}\{\log p(\mathcal{Z}, \mathcal{X}|\theta)|\mathcal{X}, \theta^{\text{old}}\} \\
&= \mathbb{E}_{z\sim P(z|\mathbf{x})}\{\log p(\mathcal{Z}, \mathcal{X}|\theta)|\mathcal{X}, \theta^{\text{old}}\} \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} P(z_{nk}=1|\mathbf{x}_n, \theta^{\text{old}}) \log p(\mathbf{x}_n, z_{nk}=1|\theta) \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma(z_{nk}) \log p(\mathbf{x}_n, z_{nk}=1|\theta) \qquad (5) \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma(z_{nk}) \log p(\mathbf{x}_n|z_{nk}=1, \theta) P(z_{nk}=1|\theta) \\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma(z_{nk}) \log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) w_k
\end{aligned}
$$

EM Algorithm for GMM

- Then, we maximize $Q(\theta|\theta^{\text{old}})$ with respect to θ by setting $\frac{\partial Q(\theta|\theta^{\text{old}})}{\partial \theta} = 0$ to obtain (see Tutorial):

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

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

$$w_k = \frac{1}{N} \sum_{n=1}^{N} \gamma(z_{nk})$$

- This constitutes the M-step.

EM Algorithm for GMM

- In practice, we compute the following sufficient statistics:

$$\text{0th-order: } N_k = \sum_{n=1}^{N} \gamma(z_{nk}) \tag{6}$$

$$\text{1st-order: } \mathbf{f}_k = \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n \tag{7}$$

$$\text{2nd-order: } \mathbf{S}_k = \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_t \mathbf{x}_t^{\mathsf{T}}, \tag{8}$$

where $k = 1, \ldots, K$.

EM Algorithm for GMM

- The model parameters are then updated as follows:

$$\boldsymbol{\mu}_k = \frac{1}{N_k}\mathbf{f}_k \tag{9}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k}\mathbf{S}_k - \boldsymbol{\mu}_k\boldsymbol{\mu}_k^{\mathsf{T}} \tag{10}$$

$$w_k = \frac{1}{N}N_k. \tag{11}$$

where $k = 1, \ldots, K$.

EM Algorithm for GMM

- In summary, the EM algorithm iteratively performs the following:
  - **Initialization**: Randomly select K samples from $\mathcal{X}$ and assign them to $\{\boldsymbol{\mu}_k\}_{k=1}^K$. Set $w_k = \frac{1}{K}$ and $\boldsymbol{\Sigma}_k = \mathbf{I}$, where $k = 1, \ldots, K$.
  - **E-Step**: Find the posterior distribution of the latent (unobserved) variables, given the observed data and the current estimate of the parameters;
  - **M-Step**: Re-estimates the parameters to maximize the likelihood of the observed data, under the assumption that the distribution found in the E-step is correct.
- The iterative process guarantees to increases the true likelihood or leaves it unchanged (if a local maximum has already been reached).

The EM Algorithm

- The EM algorithm is an ideal candidate for determining the parameters of a GMM.
- EM is applicable to the problems where the observable data provide only partial information or where some data are "missing".
- Each EM iteration is composed of two steps—Estimation (E) and Maximization (M). The M-step maximizes a likelihood function that is further refined in each iteration by the E-step.
- Animations:
  - http://davpinto.com/ml-simulations/#expectation-maximization-algorithm
  - https://www.youtube.com/watch?v=v-pq8VCQk4M

GMM: A Numerical Example

- This example uses the following data as the observed data.

- Assume that when EM begins,

$$\begin{aligned} \theta^{\text{old}} &= \{w_1, \{\mu_1, \sigma_1\}, w_2, \{\mu_2, \sigma_2\}\} \\ &= \{0.5, \{0, 1\}, 0.5, \{9, 1\}\}. \end{aligned}$$

- Therefore, one has

$$\begin{aligned} \gamma(z_{n1}) &= \frac{\frac{w_1}{\sigma_1} e^{-\frac{1}{2}(x_n-\mu_1)^2/\sigma_1^2}}{\sum_{k=1}^{2} \frac{w_k}{\sigma_k} e^{-\frac{1}{2}(x_n-\mu_k)^2/\sigma_k^2}} \\ &= \frac{e^{-\frac{1}{2}{x_n}^2}}{e^{-\frac{1}{2}{x_n}^2} + e^{-\frac{1}{2}(x_n-9)^2}} \end{aligned} \tag{12}$$

GMM: A Numerical Example

Pattern Index (t)	Pattern (x_n)	$\gamma(z_{n1})$	$\gamma(z_{n2})$
1	1	1	0
2	2	1	0
3	3	1	0
4	4	1	0
5	6	0	1
6	7	0	1
7	8	0	1

Iteration	$Q(\theta\|\theta^{\text{old}})$	μ_1	σ_1^2	μ_2	σ_2^2
0	-∞	0	1	9	1
1	-43.71	2.50	1.25	6.99	0.70
2	-25.11	2.51	1.29	7.00	0.68
3	-25.11	2.51	1.30	7.00	0.67
4	-25.10	2.52	1.30	7.00	0.67
5	-25.10	2.52	1.30	7.00	0.67

The E- and M-Steps

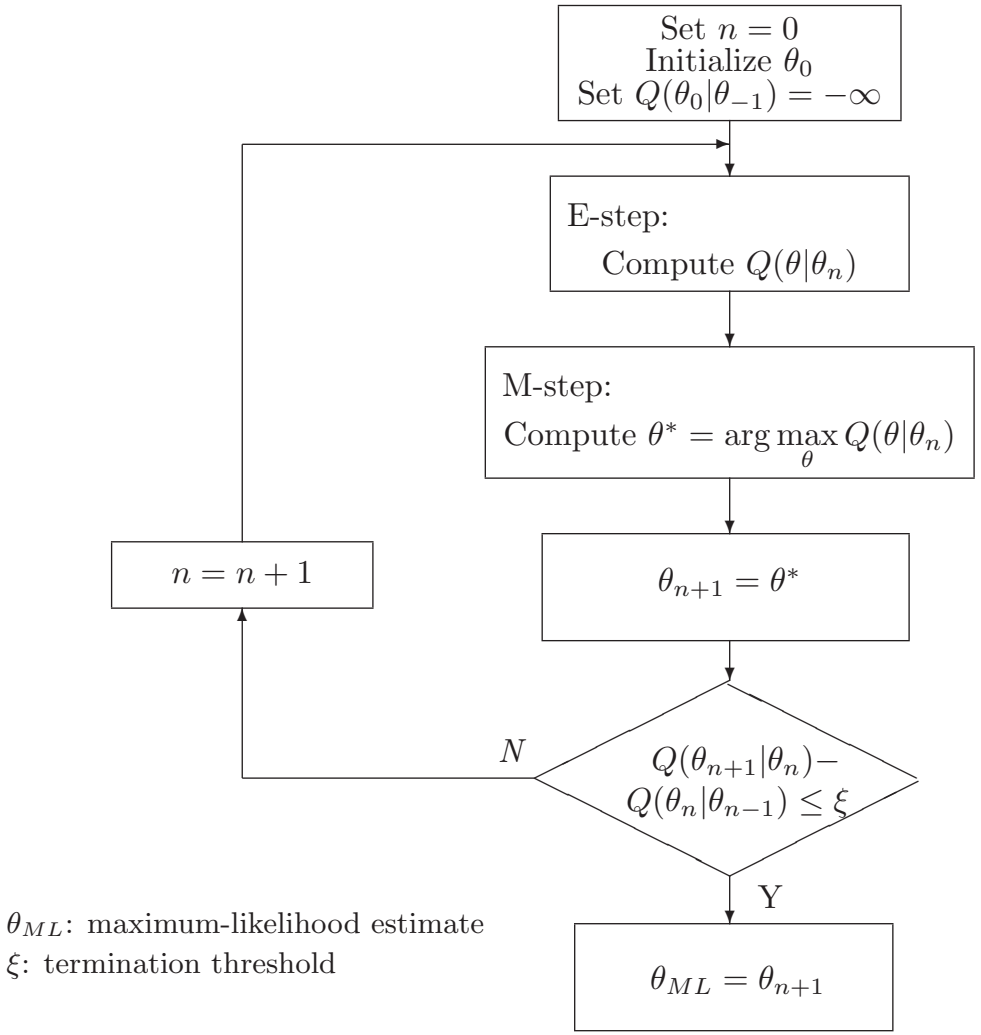
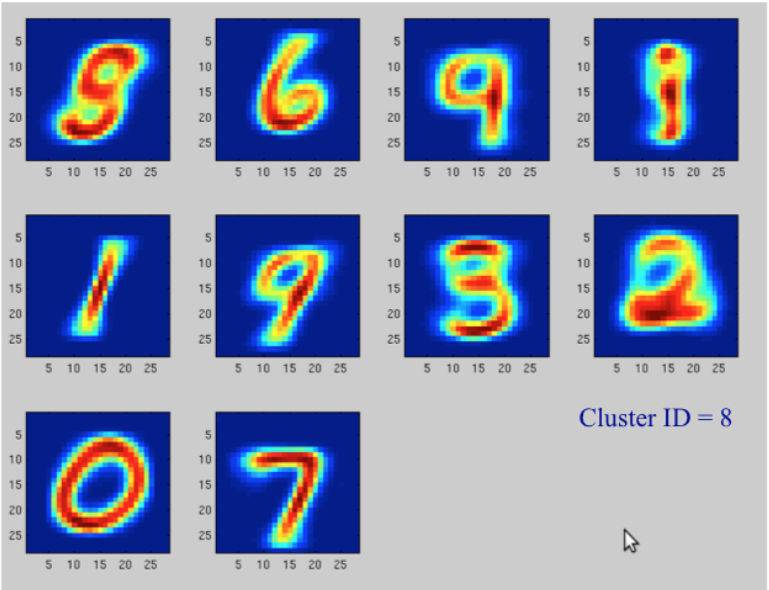
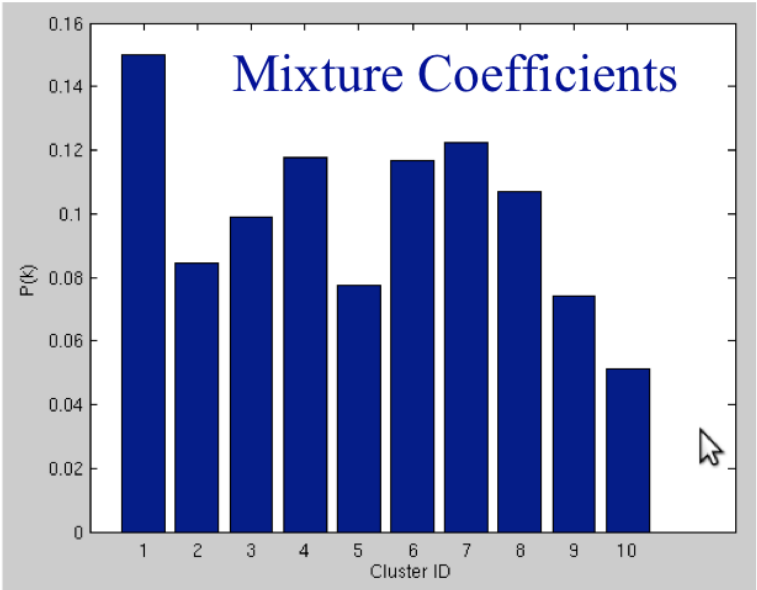


Figure: The flow of the EM algorithm.

Example Applications of GMM

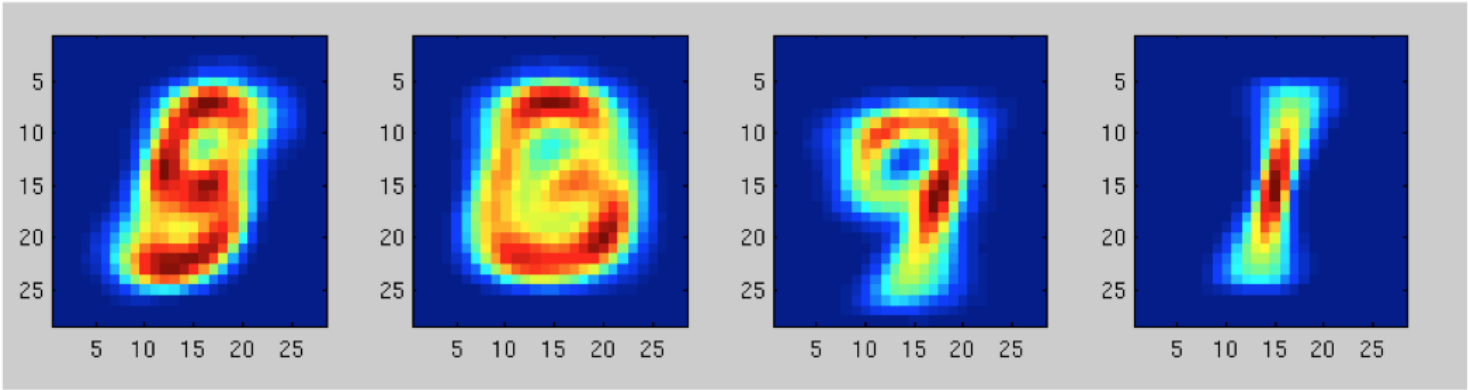
- GMM Clustering of handwritten digits with $K = 10$



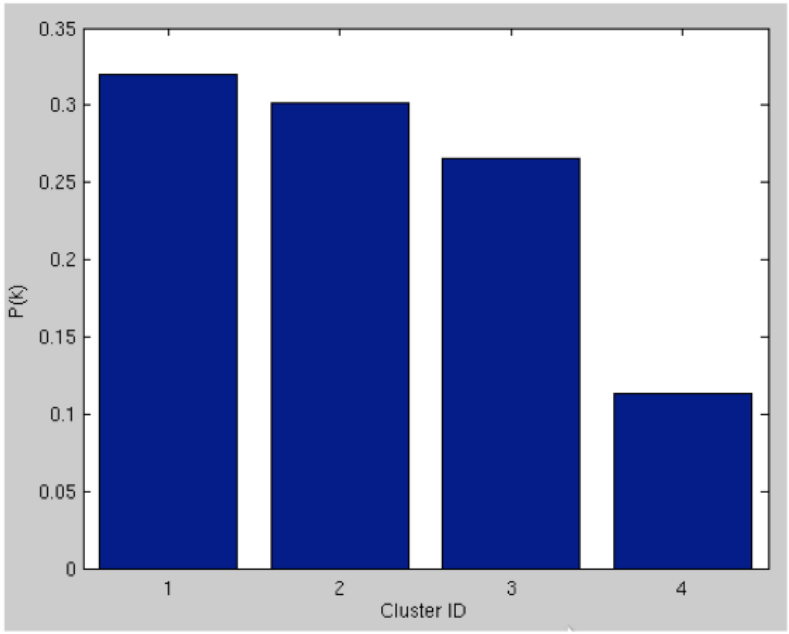


Example Applications of GMM

- GMM Clustering of handwritten digits with $K = 4$



Mixture Coefficients



Example Applications of Clustering

- DNN for Face Clustering
- https://github.com/durgeshtrivedi/imagecluster