732A99/TDDE01 Machine Learning

Lecture 1b Block 2: Gaussian Mixture Models

Jose M. Peña
IDA, Linköping University, Sweden

Contents and Literature

- Content
  - Gaussian Mixture Models
  - Supervised Learning
  - Semi-Supervised Learning
  - Unsupervised Learning
  - k-Means Algorithm
  - Summary
- Literature
  - Lindholm, A., Wahlström, N., Lindsten, F. and Schön, T. B. *Machine Learning - A First Course for Engineers and Scientists*. Cambridge University Press, 2022. Chapter 10-10.2.

Gaussian Mixture Models (GMMs)

- Most of the techniques covered in this course aim to learn a discriminative or conditional model, i.e. a model of $p(y|\boldsymbol{x})$.

Gaussian Mixture Models (GMMs)

- Most of the techniques covered in this course aim to learn a discriminative or conditional model, i.e. a model of $p(y|\boldsymbol{x})$.
- Generative modeling aims to learn a model of $p(\boldsymbol{x}, y)$.

Gaussian Mixture Models (GMMs)

- Most of the techniques covered in this course aim to learn a discriminative or conditional model, i.e. a model of $p(y|\boldsymbol{x})$.
- Generative modeling aims to learn a model of $p(\boldsymbol{x}, y)$.
- The Gaussian mixture model (GMM) is a generative model that assumes that y is categorical and $p(\boldsymbol{x}|y)$ are Gaussian distributions:

Gaussian Mixture Models (GMMs)

- Most of the techniques covered in this course aim to learn a discriminative or conditional model, i.e. a model of $p(y|\boldsymbol{x})$.
- Generative modeling aims to learn a model of $p(\boldsymbol{x}, y)$.
- The Gaussian mixture model (GMM) is a generative model that assumes that y is categorical and $p(\boldsymbol{x}|y)$ are Gaussian distributions:

$$p(\mathbf{x}, y) = p(\mathbf{x} \mid y)p(y), \tag{10.1a}$$

$$\begin{aligned} p(y = 1) &= \pi_1, \\ &\vdots \\ p(y = M) &= \pi_M. \end{aligned} \tag{10.1b}$$

$$p(\mathbf{x} \mid y) = \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y\right), \tag{10.1c}$$

Gaussian Mixture Models (GMMs)

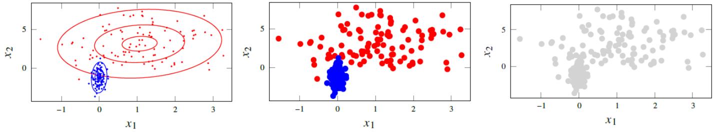
- Most of the techniques covered in this course aim to learn a discriminative or conditional model, i.e. a model of $p(y|\boldsymbol{x})$.
- Generative modeling aims to learn a model of $p(\boldsymbol{x}, y)$.
- The Gaussian mixture model (GMM) is a generative model that assumes that y is categorical and $p(\boldsymbol{x}|y)$ are Gaussian distributions:

$$p(\mathbf{x}, y) = p(\mathbf{x} \mid y)p(y), \tag{10.1a}$$

$$\begin{aligned} p(y = 1) &= \pi_1, \\ &\vdots \\ p(y = M) &= \pi_M. \end{aligned} \tag{10.1b}$$

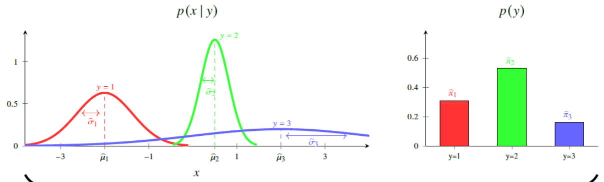
$$p(\mathbf{x} \mid y) = \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y\right), \tag{10.1c}$$

- Note that $p(\boldsymbol{x}) = \sum_{m=1}^{M} p(\boldsymbol{x}|y = m)p(y = m)$ and, thus, $p(\boldsymbol{x})$ is a mixture of Gaussian distributions, hence the name of the model.

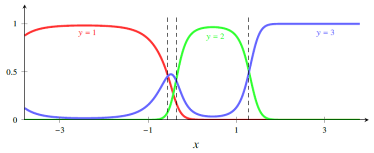


GMMs

- Note that generative models like GMMs can be turned into discriminative ones.

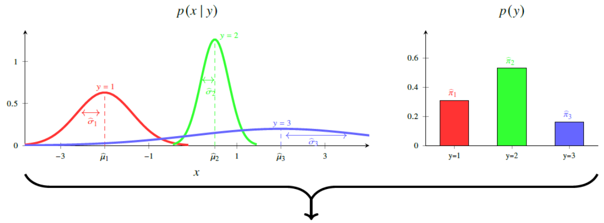


$$p(y \mid \mathbf{x}) = \frac{p(\mathbf{x}, y)}{p(\mathbf{x})} = \frac{p(\mathbf{x}, y)}{\sum_{j=1}^{M} p(\mathbf{x}, y = j)}$$

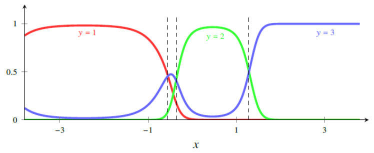


GMMs

- Note that generative models like GMMs can be turned into discriminative ones.

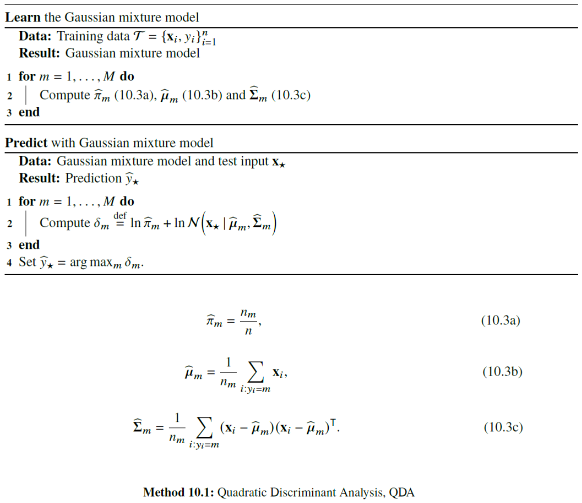


$$p(y \mid \mathbf{x}) = \frac{p(\mathbf{x}, y)}{p(\mathbf{x})} = \frac{p(\mathbf{x}, y)}{\sum_{j=1}^{M} p(\mathbf{x}, y = j)}$$



- Generative models also describe $p(\boldsymbol{x})$ and, thus, they provide a richer description of the data generating process. The downside is that they require additional assumptions on $p(\boldsymbol{x})$.

Supervised Learning of GMMs



Learn the Gaussian mixture model

```
Data: Training data $\mathcal{T} = \{\mathbf{x}_i, y_i\}_{i=1}^n$
Result: Gaussian mixture model
1 for $m = 1, \ldots, M$ do
2     Compute $\widehat{\pi}_m$ (10.3a), $\widehat{\boldsymbol{\mu}}_m$ (10.3b) and $\widehat{\boldsymbol{\Sigma}}_m$ (10.3c)
3 end
```

Predict with Gaussian mixture model

```
Data: Gaussian mixture model and test input $\mathbf{x}_\star$
Result: Prediction $\widehat{y}_\star$
1 for $m = 1, \ldots, M$ do
2     Compute $\delta_m \overset{\text{def}}{=} \ln \widehat{\pi}_m + \ln \mathcal{N}\left(\mathbf{x}_\star \mid \widehat{\boldsymbol{\mu}}_m, \widehat{\boldsymbol{\Sigma}}_m\right)$
3 end
4 Set $\widehat{y}_\star = \arg\max_m \delta_m$.
```

$$\widehat{\pi}_m = \frac{n_m}{n}, \tag{10.3a}$$

$$\widehat{\boldsymbol{\mu}}_m = \frac{1}{n_m} \sum_{i:y_i=m} \mathbf{x}_i, \tag{10.3b}$$

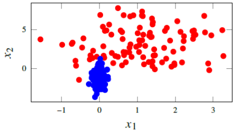
$$\widehat{\boldsymbol{\Sigma}}_m = \frac{1}{n_m} \sum_{i:y_i=m} (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_m)(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_m)^{\mathsf{T}}. \tag{10.3c}$$

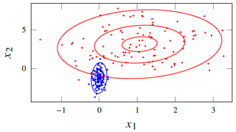
Method 10.1: Quadratic Discriminant Analysis, QDA

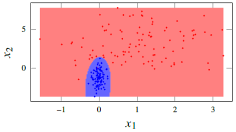
- Step 2 (learn): Equations 10.3 are obtained by the **maximum likelihood method**. Note that the parameter estimates have **closed-form expressions**, regardless of whether the training data come from a GMM or not.
- Step 2 (predict): The generative model gets turned into a discriminative one simply by noting that $p(y|\boldsymbol{x}) \propto p(\boldsymbol{x}, y) = p(\boldsymbol{x}|y)p(y)$.

Supervised Learning of GMMs

- Step 2 (predict): The logarithm of the Gaussian probability distribution is quadratic in $\boldsymbol{x}$, which implies that the decision boundary is also quadratic in $\boldsymbol{x}$, hence the name of the method.





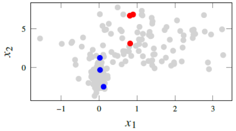


Semi-Supervised Learning of GMMs

- So far, we have considered the supervised setting in which every training point consisted of a pair input-output values $(\boldsymbol{x}, y)$.

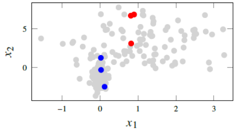
Semi-Supervised Learning of GMMs

- So far, we have considered the supervised setting in which every training point consisted of a pair input-output values $(\boldsymbol{x}, y)$.
- In the semi-supervised setting, some training points are unlabeled, i.e. the output value is missing. This is relevant in many fields where collecting input values is easier than annotating them, e.g. the bottleneck in medical diagnosis is that a domain expert has to annotate the scanner images.



Semi-Supervised Learning of GMMs

- So far, we have considered the supervised setting in which every training point consisted of a pair input-output values $(\boldsymbol{x}, y)$.
- In the semi-supervised setting, some training points are unlabeled, i.e. the output value is missing. This is relevant in many fields where collecting input values is easier than annotating them, e.g. the bottleneck in medical diagnosis is that a domain expert has to annotate the scanner images.



- The following procedure converges to a **local maximum** of the likelihood function of the training data:

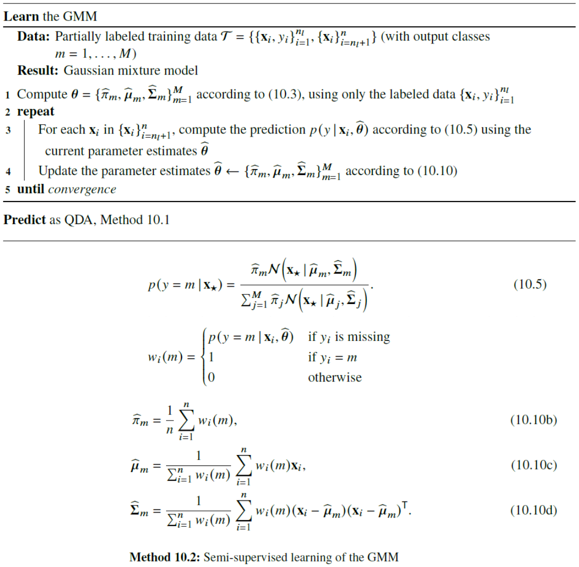
(i) Learn the GMM from the n_l labeled input-output pairs $\{\mathbf{x}_i, y_i\}_{i=1}^{n_l}$,

(ii) Use the GMM to predict (as a QDA classifier) the missing outputs to $\{\mathbf{x}_i\}_{i=n_l+1}^{n}$,

(iii) Update the GMM using also the predicted outputs from step (ii),

and then repeat step (ii) and (iii) until convergence.

Semi-Supervised Learning of GMMs



Learn the GMM

```
Data: Partially labeled training data $\mathcal{T} = \{\{\mathbf{x}_i, y_i\}_{i=1}^{n_l}, \{\mathbf{x}_i\}_{i=n_l+1}^{n}\}$ (with output classes
      $m = 1, \ldots, M$)
Result: Gaussian mixture model
1 Compute $\boldsymbol{\theta} = \{\widehat{\pi}_m, \widehat{\boldsymbol{\mu}}_m, \widehat{\boldsymbol{\Sigma}}_m\}_{m=1}^{M}$ according to (10.3), using only the labeled data $\{\mathbf{x}_i, y_i\}_{i=1}^{n_l}$
2 repeat
3     For each $\mathbf{x}_i$ in $\{\mathbf{x}_i\}_{i=n_l+1}^{n}$, compute the prediction $p(y \mid \mathbf{x}_i, \widehat{\boldsymbol{\theta}})$ according to (10.5) using the
        current parameter estimates $\widehat{\boldsymbol{\theta}}$
4     Update the parameter estimates $\widehat{\boldsymbol{\theta}} \leftarrow \{\widehat{\pi}_m, \widehat{\boldsymbol{\mu}}_m, \widehat{\boldsymbol{\Sigma}}_m\}_{m=1}^{M}$ according to (10.10)
5 until convergence
```

Predict as QDA, Method 10.1

$$p(y = m \mid \mathbf{x}_\star) = \frac{\widehat{\pi}_m \mathcal{N}\left(\mathbf{x}_\star \mid \widehat{\boldsymbol{\mu}}_m, \widehat{\boldsymbol{\Sigma}}_m\right)}{\sum_{j=1}^{M} \widehat{\pi}_j \mathcal{N}\left(\mathbf{x}_\star \mid \widehat{\boldsymbol{\mu}}_j, \widehat{\boldsymbol{\Sigma}}_j\right)}. \tag{10.5}$$

$$w_i(m) = \begin{cases} p(y = m \mid \mathbf{x}_i, \widehat{\boldsymbol{\theta}}) & \text{if } y_i \text{ is missing} \\ 1 & \text{if } y_i = m \\ 0 & \text{otherwise} \end{cases}$$

$$\widehat{\pi}_m = \frac{1}{n} \sum_{i=1}^{n} w_i(m), \tag{10.10b}$$

$$\widehat{\boldsymbol{\mu}}_m = \frac{1}{\sum_{i=1}^{n} w_i(m)} \sum_{i=1}^{n} w_i(m) \mathbf{x}_i, \tag{10.10c}$$

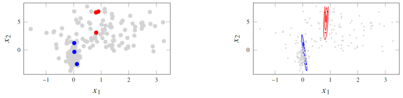
$$\widehat{\boldsymbol{\Sigma}}_m = \frac{1}{\sum_{i=1}^{n} w_i(m)} \sum_{i=1}^{n} w_i(m) (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_m)(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_m)^\mathsf{T}. \tag{10.10d}$$

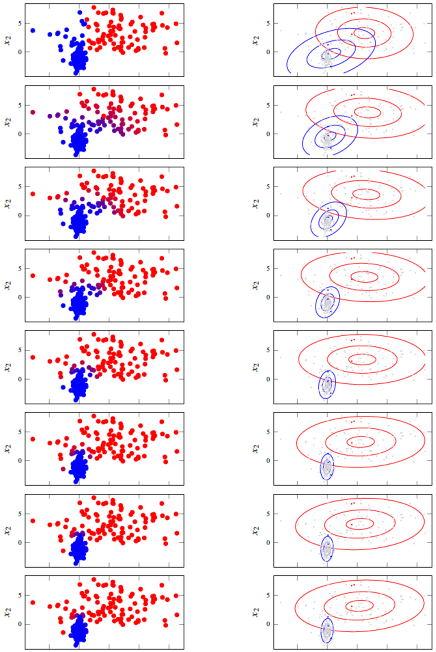
Method 10.2: Semi-supervised learning of the GMM

- Equations 10.10: Note that the unlabeled training points contribute proportionally to the probability estimate of belonging to the class.

Semi-Supervised Learning of GMMs

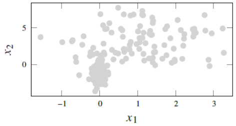
Initialization with only labeled points





Unsupervised Learning of GMMs

- In the unsupervised setting, all the training points are unlabeled. The number of different labels (a.k.a. clusters) is **unknown**. So, the unsupervised learning (a.k.a. clustering) is an exploratory task, aimed at explaining the data in terms of interpretable clusters.



Unsupervised Learning of GMMs

Learn the GMM

```
   Data: Unlabeled training data $\mathcal{T} = \{\mathbf{x}_i\}_{i=1}^n$, number of clusters $M$.
   Result: Gaussian mixture model
1 Initialize $\widehat{\boldsymbol{\theta}} = \{\widehat{\pi}_m, \widehat{\boldsymbol{\mu}}_m, \widehat{\boldsymbol{\Sigma}}_m\}_{m=1}^M$
2 repeat
3     For each $\mathbf{x}_i$ in $\{\mathbf{x}_i\}_{i=1}^n$, compute the prediction $p(y \mid \mathbf{x}_i, \widehat{\boldsymbol{\theta}})$ according to (10.5) using the current
        parameter estimates $\widehat{\boldsymbol{\theta}}$.
4     Update the parameter estimates $\widehat{\boldsymbol{\theta}} \leftarrow \{\widehat{\pi}_m, \widehat{\boldsymbol{\mu}}_m, \widehat{\boldsymbol{\Sigma}}_m\}_{m=1}^M$ according to (10.16)
5 until convergence
```

Predict as QDA, Method 10.1

$$p(y = m \mid \mathbf{x}_\star) = \frac{\widehat{\pi}_m \mathcal{N}\left(\mathbf{x}_\star \mid \widehat{\boldsymbol{\mu}}_m, \widehat{\boldsymbol{\Sigma}}_m\right)}{\sum_{j=1}^M \widehat{\pi}_j \mathcal{N}\left(\mathbf{x}_\star \mid \widehat{\boldsymbol{\mu}}_j, \widehat{\boldsymbol{\Sigma}}_j\right)}. \tag{10.5}$$

$$\widehat{\pi}_m = \frac{1}{n}\sum_{i=1}^n w_i(m), \tag{10.16a}$$

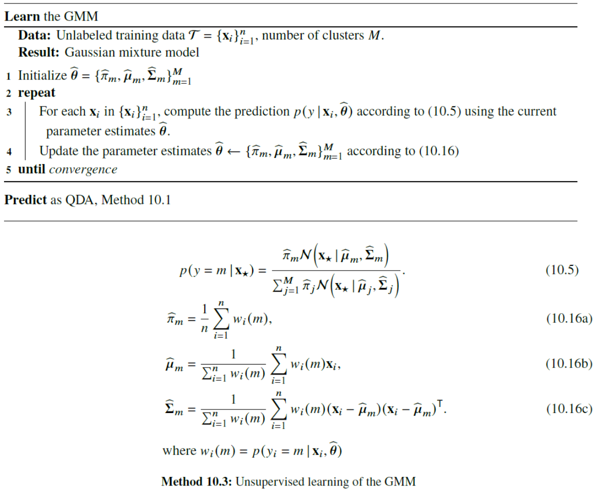
$$\widehat{\boldsymbol{\mu}}_m = \frac{1}{\sum_{i=1}^n w_i(m)}\sum_{i=1}^n w_i(m)\mathbf{x}_i, \tag{10.16b}$$

$$\widehat{\boldsymbol{\Sigma}}_m = \frac{1}{\sum_{i=1}^n w_i(m)}\sum_{i=1}^n w_i(m)(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_m)(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_m)^\mathsf{T}. \tag{10.16c}$$

where $w_i(m) = p(y_i = m \mid \mathbf{x}_i, \widehat{\boldsymbol{\theta}})$

Method 10.3: Unsupervised learning of the GMM

Unsupervised Learning of GMMs



Learn the GMM

```
Data: Unlabeled training data $\mathcal{T} = \{\mathbf{x}_i\}_{i=1}^n$, number of clusters $M$.
Result: Gaussian mixture model
1 Initialize $\widehat{\boldsymbol{\theta}} = \{\widehat{\pi}_m, \widehat{\boldsymbol{\mu}}_m, \widehat{\boldsymbol{\Sigma}}_m\}_{m=1}^M$
2 repeat
3     For each $\mathbf{x}_i$ in $\{\mathbf{x}_i\}_{i=1}^n$, compute the prediction $p(y \mid \mathbf{x}_i, \widehat{\boldsymbol{\theta}})$ according to (10.5) using the current
        parameter estimates $\widehat{\boldsymbol{\theta}}$.
4     Update the parameter estimates $\widehat{\boldsymbol{\theta}} \leftarrow \{\widehat{\pi}_m, \widehat{\boldsymbol{\mu}}_m, \widehat{\boldsymbol{\Sigma}}_m\}_{m=1}^M$ according to (10.16)
5 until convergence
```

Predict as QDA, Method 10.1

$$p(y = m \mid \mathbf{x}_\star) = \frac{\widehat{\pi}_m \mathcal{N}\left(\mathbf{x}_\star \mid \widehat{\boldsymbol{\mu}}_m, \widehat{\boldsymbol{\Sigma}}_m\right)}{\sum_{j=1}^M \widehat{\pi}_j \mathcal{N}\left(\mathbf{x}_\star \mid \widehat{\boldsymbol{\mu}}_j, \widehat{\boldsymbol{\Sigma}}_j\right)}. \tag{10.5}$$

$$\widehat{\pi}_m = \frac{1}{n} \sum_{i=1}^n w_i(m), \tag{10.16a}$$

$$\widehat{\boldsymbol{\mu}}_m = \frac{1}{\sum_{i=1}^n w_i(m)} \sum_{i=1}^n w_i(m) \mathbf{x}_i, \tag{10.16b}$$

$$\widehat{\boldsymbol{\Sigma}}_m = \frac{1}{\sum_{i=1}^n w_i(m)} \sum_{i=1}^n w_i(m)(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_m)(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_m)^\mathsf{T}. \tag{10.16c}$$

where $w_i(m) = p(y_i = m \mid \mathbf{x}_i, \widehat{\boldsymbol{\theta}})$

Method 10.3: Unsupervised learning of the GMM

- The algorithm above is also known as EM algorithm: Step 3 is called Expectation, and step 4 is called Maximization.

Unsupervised Learning of GMMs

Learn the GMM

```
Data: Unlabeled training data $\mathcal{T} = \{\mathbf{x}_i\}_{i=1}^n$, number of clusters $M$.
Result: Gaussian mixture model
1 Initialize $\widehat{\boldsymbol{\theta}} = \{\widehat{\pi}_m, \widehat{\boldsymbol{\mu}}_m, \widehat{\boldsymbol{\Sigma}}_m\}_{m=1}^M$
2 repeat
3     For each $\mathbf{x}_i$ in $\{\mathbf{x}_i\}_{i=1}^n$, compute the prediction $p(y \,|\, \mathbf{x}_i, \widehat{\boldsymbol{\theta}})$ according to (10.5) using the current parameter estimates $\widehat{\boldsymbol{\theta}}$.
4     Update the parameter estimates $\widehat{\boldsymbol{\theta}} \leftarrow \{\widehat{\pi}_m, \widehat{\boldsymbol{\mu}}_m, \widehat{\boldsymbol{\Sigma}}_m\}_{m=1}^M$ according to (10.16)
5 until convergence
```

Predict as QDA, Method 10.1

$$p(y = m \,|\, \mathbf{x}_\star) = \frac{\widehat{\pi}_m \mathcal{N}\left(\mathbf{x}_\star \,|\, \widehat{\boldsymbol{\mu}}_m, \widehat{\boldsymbol{\Sigma}}_m\right)}{\sum_{j=1}^M \widehat{\pi}_j \mathcal{N}\left(\mathbf{x}_\star \,|\, \widehat{\boldsymbol{\mu}}_j, \widehat{\boldsymbol{\Sigma}}_j\right)}. \tag{10.5}$$

$$\widehat{\pi}_m = \frac{1}{n}\sum_{i=1}^n w_i(m), \tag{10.16a}$$

$$\widehat{\boldsymbol{\mu}}_m = \frac{1}{\sum_{i=1}^n w_i(m)} \sum_{i=1}^n w_i(m)\mathbf{x}_i, \tag{10.16b}$$

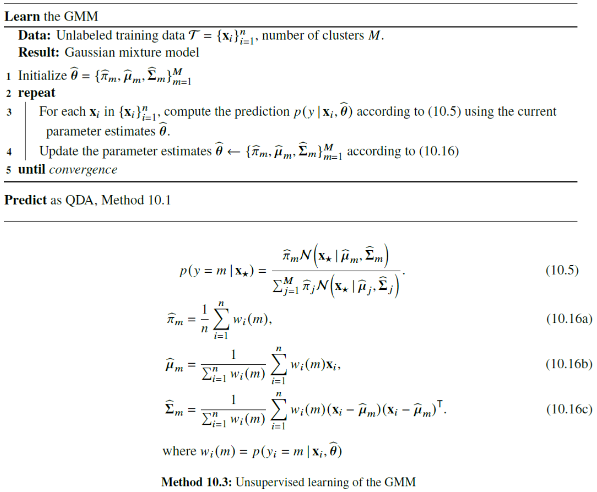
$$\widehat{\boldsymbol{\Sigma}}_m = \frac{1}{\sum_{i=1}^n w_i(m)} \sum_{i=1}^n w_i(m)(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_m)(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_m)^\mathsf{T}. \tag{10.16c}$$

where $w_i(m) = p(y_i = m \,|\, \mathbf{x}_i, \widehat{\boldsymbol{\theta}})$

Method 10.3: Unsupervised learning of the GMM

- The algorithm above is also known as EM algorithm: Step 3 is called Expectation, and step 4 is called Maximization.
- It converges to a **local maximum** of the likelihood function of the training data.

Unsupervised Learning of GMMs



Learn the GMM

```
Data: Unlabeled training data $\mathcal{T} = \{\mathbf{x}_i\}_{i=1}^n$, number of clusters $M$.
Result: Gaussian mixture model
1 Initialize $\widehat{\boldsymbol{\theta}} = \{\widehat{\pi}_m, \widehat{\boldsymbol{\mu}}_m, \widehat{\boldsymbol{\Sigma}}_m\}_{m=1}^M$
2 repeat
3     For each $\mathbf{x}_i$ in $\{\mathbf{x}_i\}_{i=1}^n$, compute the prediction $p(y \mid \mathbf{x}_i, \widehat{\boldsymbol{\theta}})$ according to (10.5) using the current
      parameter estimates $\widehat{\boldsymbol{\theta}}$.
4     Update the parameter estimates $\widehat{\boldsymbol{\theta}} \leftarrow \{\widehat{\pi}_m, \widehat{\boldsymbol{\mu}}_m, \widehat{\boldsymbol{\Sigma}}_m\}_{m=1}^M$ according to (10.16)
5 until convergence
```

Predict as QDA, Method 10.1

$$p(y = m \mid \mathbf{x}_\star) = \frac{\widehat{\pi}_m \mathcal{N}\left(\mathbf{x}_\star \mid \widehat{\boldsymbol{\mu}}_m, \widehat{\boldsymbol{\Sigma}}_m\right)}{\sum_{j=1}^M \widehat{\pi}_j \mathcal{N}\left(\mathbf{x}_\star \mid \widehat{\boldsymbol{\mu}}_j, \widehat{\boldsymbol{\Sigma}}_j\right)}. \tag{10.5}$$

$$\widehat{\pi}_m = \frac{1}{n} \sum_{i=1}^n w_i(m), \tag{10.16a}$$

$$\widehat{\boldsymbol{\mu}}_m = \frac{1}{\sum_{i=1}^n w_i(m)} \sum_{i=1}^n w_i(m) \mathbf{x}_i, \tag{10.16b}$$

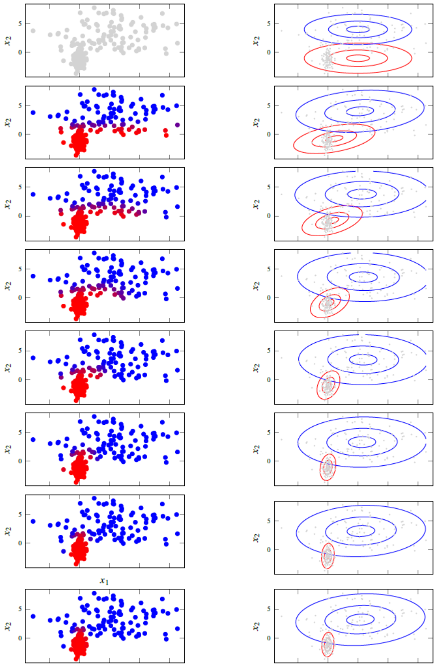
$$\widehat{\boldsymbol{\Sigma}}_m = \frac{1}{\sum_{i=1}^n w_i(m)} \sum_{i=1}^n w_i(m) (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_m)(\mathbf{x}_i - \widehat{\boldsymbol{\mu}}_m)^\mathsf{T}. \tag{10.16c}$$

where $w_i(m) = p(y_i = m \mid \mathbf{x}_i, \widehat{\boldsymbol{\theta}})$

Method 10.3: Unsupervised learning of the GMM

- The algorithm above is also known as EM algorithm: Step 3 is called Expectation, and step 4 is called Maximization.
- It converges to a **local maximum** of the likelihood function of the training data.
- The number of clusters is a user-defined parameter. Hold-out and cross-validation can be of help. Note that the cluster labels are arbitrary, i.e. all the permutations of the cluster labels have the same likelihood.

Unsupervised Learning of GMMs



k-Means Algorithm

- Recall that

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

k-Means Algorithm

- Recall that

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

- Consider $\boldsymbol{\Sigma}_m = \epsilon \boldsymbol{I}$ where ϵ is a positive scalar and $\boldsymbol{I}$ is the identity matrix. Then,

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

$$p(y=m|\boldsymbol{x}) = \frac{\pi_m p(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{j=1}^{M} \pi_j p(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = \frac{\pi_m \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{j=1}^{M} \pi_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = \frac{\pi_m \exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_m||^2)}{\sum_{j=1}^{M} \pi_j \exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_j||^2)}$$

k-Means Algorithm

- Recall that

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

- Consider $\boldsymbol{\Sigma}_m = \epsilon \boldsymbol{I}$ where ϵ is a positive scalar and $\boldsymbol{I}$ is the identity matrix. Then,

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

$$p(y=m|\boldsymbol{x}) = \frac{\pi_m p(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{j=1}^M \pi_j p(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = \frac{\pi_m \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{j=1}^M \pi_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = \frac{\pi_m \exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_m||^2)}{\sum_{j=1}^M \pi_j \exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_j||^2)}$$

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

k-Means Algorithm

- Recall that

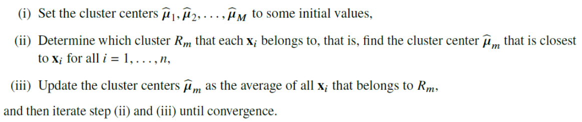
$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) = (2\pi)^{-q/2} det(\boldsymbol{\Sigma}_m)^{-1/2} \exp(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_m)^T \boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_m))$$

- Consider $\boldsymbol{\Sigma}_m = \epsilon \boldsymbol{I}$ where ϵ is a positive scalar and $\boldsymbol{I}$ is the identity matrix. Then,

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

$$p(y=m|\boldsymbol{x}) = \frac{\pi_m p(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{j=1}^M \pi_j p(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = \frac{\pi_m \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{j=1}^M \pi_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = \frac{\pi_m \exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_m||^2)}{\sum_{j=1}^M \pi_j \exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_j||^2)}$$

- As $\epsilon \to 0$, the smaller $||\boldsymbol{x}-\boldsymbol{\mu}_m||^2$ the slower $\exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_m||^2)$ goes to 0.
- As $\epsilon \to 0$, training points are hard-assigned (i.e. with probability 1) to the cluster with closest mean. This version of the EM algorithm is known as *k*-means algorithm:



(i) Set the cluster centers $\widehat{\mu}_1, \widehat{\mu}_2, \ldots, \widehat{\mu}_M$ to some initial values,

(ii) Determine which cluster R_m that each $\mathbf{x}_i$ belongs to, that is, find the cluster center $\widehat{\mu}_m$ that is closest to $\mathbf{x}_i$ for all $i = 1, \ldots, n$,

(iii) Update the cluster centers $\widehat{\mu}_m$ as the average of all $\mathbf{x}_i$ that belongs to R_m,

and then iterate step (ii) and (iii) until convergence.

k-Means Algorithm

- Recall that

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

- Consider $\boldsymbol{\Sigma}_m = \epsilon \boldsymbol{I}$ where ϵ is a positive scalar and $\boldsymbol{I}$ is the identity matrix. Then,

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

$$p(y=m|\boldsymbol{x}) = \frac{\pi_m p(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{j=1}^M \pi_j p(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = \frac{\pi_m \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{j=1}^M \pi_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = \frac{\pi_m \exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_m||^2)}{\sum_{j=1}^M \pi_j \exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_j||^2)}$$

- As $\epsilon \to 0$, the smaller $||\boldsymbol{x}-\boldsymbol{\mu}_m||^2$ the slower $\exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_m||^2)$ goes to 0.
- As $\epsilon \to 0$, training points are hard-assigned (i.e. with probability 1) to the cluster with closest mean. This version of the EM algorithm is known as k-means algorithm:

(i) Set the cluster centers $\widehat{\boldsymbol{\mu}}_1, \widehat{\boldsymbol{\mu}}_2, \dots, \widehat{\boldsymbol{\mu}}_M$ to some initial values,

(ii) Determine which cluster R_m that each $\mathbf{x}_i$ belongs to, that is, find the cluster center $\widehat{\boldsymbol{\mu}}_m$ that is closest to $\mathbf{x}_i$ for all $i = 1, \dots, n$,

(iii) Update the cluster centers $\widehat{\boldsymbol{\mu}}_m$ as the average of all $\mathbf{x}_i$ that belongs to R_m,

and then iterate step (ii) and (iii) until convergence.

- Note that $\{\widehat{\pi}_m, \widehat{\boldsymbol{\Sigma}}_m\}_{m=1}^M$ play no role in the k-means algorithm.

k-Means Algorithm

- Recall that

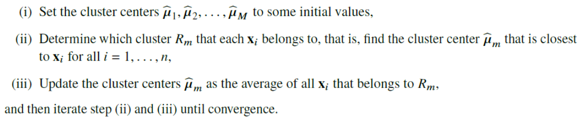
$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) = (2\pi)^{-q/2} det(\boldsymbol{\Sigma}_m)^{-1/2} \exp(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_m)^T \boldsymbol{\Sigma}_m^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_m))$$

- Consider $\boldsymbol{\Sigma}_m = \epsilon \boldsymbol{I}$ where ϵ is a positive scalar and $\boldsymbol{I}$ is the identity matrix. Then,

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

$$p(y=m|\boldsymbol{x}) = \frac{\pi_m p(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{j=1}^{M} \pi_j p(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = \frac{\pi_m \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{j=1}^{M} \pi_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = \frac{\pi_m \exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_m||^2)}{\sum_{j=1}^{M} \pi_j \exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_j||^2)}$$

- As $\epsilon \to 0$, the smaller $||\boldsymbol{x}-\boldsymbol{\mu}_m||^2$ the slower $\exp(-\frac{1}{2\epsilon}||\boldsymbol{x}-\boldsymbol{\mu}_m||^2)$ goes to 0.
- As $\epsilon \to 0$, training points are hard-assigned (i.e. with probability 1) to the cluster with closest mean. This version of the EM algorithm is known as k-means algorithm:



(i) Set the cluster centers $\widehat{\mu}_1, \widehat{\mu}_2, \ldots, \widehat{\mu}_M$ to some initial values,

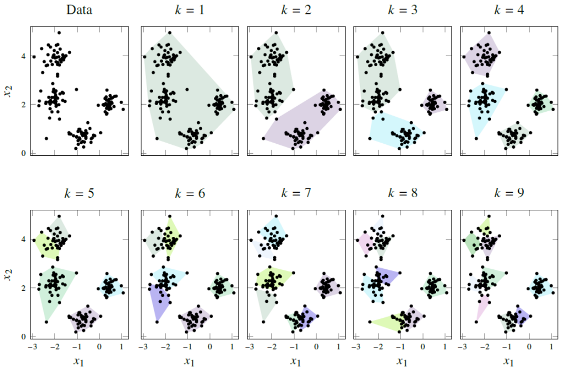
(ii) Determine which cluster R_m that each $\mathbf{x}_i$ belongs to, that is, find the cluster center $\widehat{\mu}_m$ that is closest to $\mathbf{x}_i$ for all $i = 1, \ldots, n$,

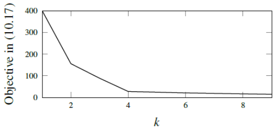
(iii) Update the cluster centers $\widehat{\mu}_m$ as the average of all $\mathbf{x}_i$ that belongs to R_m,

and then iterate step (ii) and (iii) until convergence.

- Note that $\{\widehat{\pi}_m, \widehat{\boldsymbol{\Sigma}}_m\}_{m=1}^{M}$ play no role in the k-means algorithm.
- The k-means algorithm can be used to initialize the EM algorithm.

k-Means Algorithm





$$\arg\min_{R_1,R_2,\ldots,R_M} \sum_{m=1}^{M} \frac{1}{|R_m|} \sum_{\mathbf{x},\mathbf{x}' \in R_m} \|\mathbf{x}-\mathbf{x}'\|_2^2, \qquad (10.17)$$

Figure 10.9: For selecting M in k-means we can use the so-called elbow method, which amounts to trying different values of M (the upper panels) and record the objective in (10.17) (the bottom panel). To select M, we look for a "bend' in the bottom panel. In an ideal case there is a very distinct kink, but for this particular data we could either draw the conclusion that $M = 2$ or $M = 4$ and it is up to the user to decide. Note that in this example the data has only 2 dimensions, and we can therefore show the clusters themselves and compare them visually. If the data has more than two dimensions, however, we have to select M based only on the "elbow plot" in the bottom panel.

Summary

- Gaussian Mixture Models
- Supervised Learning
- Semi-Supervised Learning
- Unsupervised Learning
- k-Means Algorithm