CS 6316 Machine Learning

Generative Models

Yangfeng Ji

Information and Language Processing Lab
Department of Computer Science
University of Virginia

Basic Definition

Data generation process

An idealized process to illustrate the relations among domain set $\mathcal{X}$, label set $\mathcal{Y}$, and the training set S

1. the probability distribution $\mathcal{D}$ over the domain set $\mathcal{X}$
2. sample an instance $\boldsymbol{x} \in \mathcal{X}$ according to $\mathcal{D}$
3. annotate it using the labeling function f as $y = f(\boldsymbol{x})$

[From Lecture 01]

Example

Here is an data generation model

$$p(\boldsymbol{x}) = \underbrace{0.6 \cdot \mathcal{N}(x; \boldsymbol{\mu}_+, \boldsymbol{\Sigma}_+)}_{y=+1} + \underbrace{0.4 \cdot \mathcal{N}(x; \boldsymbol{\mu}_-, \boldsymbol{\Sigma}_-)}_{y=-1} \tag{1}$$

with

- $\boldsymbol{\mu}_+ = [2, 0]^\mathsf{T}$
- $\boldsymbol{\Sigma}_+ = \begin{bmatrix} 1.0 & 0.8 \\ 0.8 & 2.0 \end{bmatrix}$
- $\boldsymbol{\mu}_- = [-2, 0]^\mathsf{T}$
- $\boldsymbol{\Sigma}_- = \begin{bmatrix} 2.0 & 0.6 \\ 0.6 & 1.0 \end{bmatrix}$

Example (II)

The data generation model can also be represented with the following components

$$p(y = +1) = 0.6 \tag{2}$$

$$p(y = -1) = 1 - p(y = +1) = 0.4 \tag{3}$$

$$p(\boldsymbol{x} \mid y = +1) = \mathcal{N}(x; \boldsymbol{\mu}_{+}, \boldsymbol{\Sigma}_{+}) \tag{4}$$

$$p(\boldsymbol{x} \mid y = -1) = \mathcal{N}(x; \boldsymbol{\mu}_{-}, \boldsymbol{\Sigma}_{-}) \tag{5}$$

Data Generation

The specific data generation process:
for each data point

1. Randomly select a value of $y \in \{+1, -1\}$ based on

$$p(y = +1) = 0.6 \quad p(y = -1) = 0.4 \tag{6}$$

Data Generation

The specific data generation process:
for each data point

1. Randomly select a value of $y \in \{+1, -1\}$ based on

$$p(y = +1) = 0.6 \quad p(y = -1) = 0.4 \tag{6}$$

2. Sample $\boldsymbol{x}$ from the corresponding component based on the value of y

$$p(\boldsymbol{x} \mid y) = \begin{cases} \mathcal{N}(x; \boldsymbol{\mu}_{+}, \boldsymbol{\Sigma}_{+}) & y = +1 \\ \mathcal{N}(x; \boldsymbol{\mu}_{-}, \boldsymbol{\Sigma}_{-}) & y = -1 \end{cases} \tag{7}$$

Data Generation

The specific data generation process:
for each data point

1. Randomly select a value of $y \in \{+1, -1\}$ based on

$$p(y = +1) = 0.6 \quad p(y = -1) = 0.4 \tag{6}$$

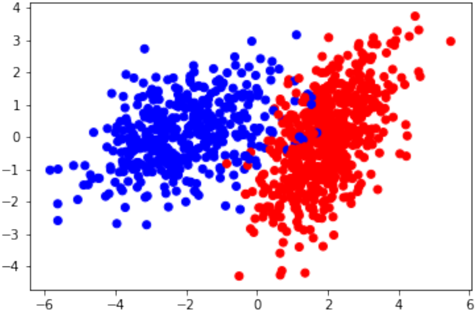
2. Sample $\boldsymbol{x}$ from the corresponding component based on the value of y

$$p(\boldsymbol{x} \mid y) = \begin{cases} \mathcal{N}(x; \boldsymbol{\mu}_{+}, \boldsymbol{\Sigma}_{+}) & y = +1 \\ \mathcal{N}(x; \boldsymbol{\mu}_{-}, \boldsymbol{\Sigma}_{-}) & y = -1 \end{cases} \tag{7}$$

3. Add $(\boldsymbol{x}, y)$ to S, go to step 1

Illustration

With $N = 1000$ samples, here is the plot



- 588 positive samples and 412 negative samples

Discriminative Models for Classification

- Discriminative models directly give predictions on the target variable (e.g., y)
- Example: logistic regression

$$p(y \mid \boldsymbol{x}) = \sigma(y\langle \boldsymbol{w}, \boldsymbol{x}\rangle) = \frac{1}{1 + e^{-y\langle \boldsymbol{w}, \boldsymbol{x}\rangle}} \tag{8}$$

where $\boldsymbol{w}$ is the model parameter

Discriminative Models for Classification

- Discriminative models directly give predictions on the target variable (e.g., y)
- Example: logistic regression

$$p(y \mid \boldsymbol{x}) = \sigma(y\langle \boldsymbol{w}, \boldsymbol{x}\rangle) = \frac{1}{1 + e^{-y\langle \boldsymbol{w}, \boldsymbol{x}\rangle}} \tag{8}$$

  where $\boldsymbol{w}$ is the model parameter
- Other examples
  - SVM with various kernels
  - Feed-forward neural network

Generative Models for Classification

- Basic idea: Building a classifier by *simulating* the data generation process

Generative Models for Classification

- Basic idea: Building a classifier by *simulating* the data generation process
- For the binary classification problem, recall the basic components of the data generation process
  - $p(y)$ where $y \in \{-1, +1\}$
  - $p(\boldsymbol{x} \mid y = +1)$ where $\boldsymbol{x} \in \mathbb{R}^d$
  - $p(\boldsymbol{x} \mid y = -1)$ where $\boldsymbol{x} \in \mathbb{R}^d$

Generative Models for Classification

- Basic idea: Building a classifier by *simulating* the data generation process
- For the binary classification problem, recall the basic components of the data generation process
  - $p(y)$ where $y \in \{-1, +1\}$
  - $p(\boldsymbol{x} \mid y = +1)$ where $\boldsymbol{x} \in \mathbb{R}^d$
  - $p(\boldsymbol{x} \mid y = -1)$ where $\boldsymbol{x} \in \mathbb{R}^d$
- Challenge in machine learning: we do not know any of them, instead we have the samples S from this distribution
  - This has always been our assumption in machine learning — we have no idea about the true data distribution

Generative Models for Classification (II)

We use a set of distribution $q(\cdot)$ to approximate the true distribution $p(\cdot)$

Data Generation Model	Generative Model
$p(y)$	$q(y)$
$p(\boldsymbol{x} \mid y = +1)$	$q(\boldsymbol{x} \mid y = +1)$
$p(\boldsymbol{x} \mid y = -1)$	$q(\boldsymbol{x} \mid y = -1)$

Learning with Generative Models

1. Define distributions for all components
2. Estimate the parameters for each component distribution

Defining Distributions

A typical way of defining distributions for generative models is based on *our understanding about the problem*

Defining Distributions

A typical way of defining distributions for generative models is based on *our understanding about the problem*

- Output domain $y \in \{+1, -1\}$: **Bernoulli** distribution

$$p(y) = \text{Bern}(y; \alpha) = \alpha^{\delta(y=+1)}(1-\alpha)^{\delta(y=-1)} \tag{9}$$

  where $\alpha \in (0, 1)$ is the parameter

Defining Distributions

A typical way of defining distributions for generative models is based on *our understanding about the problem*

- Output domain $y \in \{+1, -1\}$: **Bernoulli** distribution

$$p(y) = \mathrm{Bern}(y; \alpha) = \alpha^{\delta(y=+1)}(1-\alpha)^{\delta(y=-1)} \tag{9}$$

  where $\alpha \in (0, 1)$ is the parameter

- Input domain $\boldsymbol{x} \in \mathbb{R}^d$: **Gaussian** distribution

$$p(\boldsymbol{x} \mid y = +1) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{+}, \boldsymbol{\Sigma}_{+}) \tag{10}$$

  where $\boldsymbol{\mu}_{+}$ and $\boldsymbol{\Sigma}_{+}$ are the parameters

Defining Distributions

A typical way of defining distributions for generative models is based on *our understanding about the problem*

- Output domain $y \in \{+1, -1\}$: **Bernoulli** distribution

$$p(y) = \mathrm{Bern}(y; \alpha) = \alpha^{\delta(y=+1)}(1-\alpha)^{\delta(y=-1)} \tag{9}$$

  where $\alpha \in (0, 1)$ is the parameter

- Input domain $\boldsymbol{x} \in \mathbb{R}^d$: **Gaussian** distribution

$$p(\boldsymbol{x} \mid y = +1) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_+, \boldsymbol{\Sigma}_+) \tag{10}$$

  where $\boldsymbol{\mu}_+$ and $\boldsymbol{\Sigma}_+$ are the parameters

- Similarly, for $p(\boldsymbol{x} \mid y = -1)$

$$p(\boldsymbol{x} \mid y = -1) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_-, \boldsymbol{\Sigma}_-) \tag{11}$$

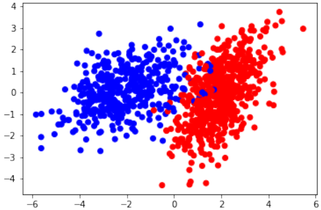
  where $\boldsymbol{\mu}_-$ and $\boldsymbol{\Sigma}_-$ are the parameters

Parameter Estimation

- The collection of the parameters

$$\boldsymbol{\theta} = \{\alpha, \boldsymbol{\mu}_{+}, \boldsymbol{\Sigma}_{+}, \boldsymbol{\mu}_{-}, \boldsymbol{\Sigma}_{-}\} \tag{12}$$

- Training data $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$

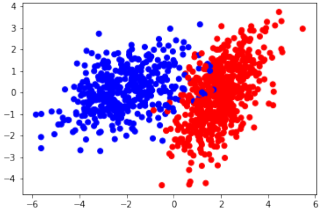


Parameter Estimation

- The collection of the parameters

$$\boldsymbol{\theta} = \{\alpha, \boldsymbol{\mu}_{+}, \boldsymbol{\Sigma}_{+}, \boldsymbol{\mu}_{-}, \boldsymbol{\Sigma}_{-}\} \tag{12}$$

- Training data $S = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$



- Learning algorithm: Maximum Likelihood Estimation (MLE)

Maximum Likelihood Estimation (MLE)

MLE defined on the whole distribution $q(\boldsymbol{x}, y)$

$$\boldsymbol{\theta} \leftarrow \underset{\boldsymbol{\theta}'}{\operatorname{argmax}} \sum_{i=1}^{m} \log q(\boldsymbol{x}_i, y_i; \boldsymbol{\theta}') \tag{13}$$

Maximum Likelihood Estimation (MLE)

MLE defined on the whole distribution $q(\boldsymbol{x}, y)$

$$\boldsymbol{\theta} \leftarrow \underset{\boldsymbol{\theta}'}{\operatorname{argmax}} \sum_{i=1}^{m} \log q(\boldsymbol{x}_i, y_i; \boldsymbol{\theta}') \tag{13}$$

Based on the chain rule of probability

$$q(\boldsymbol{x}, y; \boldsymbol{\theta}) = q(y; \alpha) q(\boldsymbol{x} \mid y; \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y), \tag{14}$$

Maximum Likelihood Estimation (MLE)

MLE defined on the whole distribution $q(\boldsymbol{x}, y)$

$$\boldsymbol{\theta} \leftarrow \operatorname*{argmax}_{\boldsymbol{\theta}'} \sum_{i=1}^{m} \log q(\boldsymbol{x}_i, y_i; \boldsymbol{\theta}') \tag{13}$$

Based on the chain rule of probability

$$q(\boldsymbol{x}, y; \boldsymbol{\theta}) = q(y; \alpha) q(\boldsymbol{x} \mid y; \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y), \tag{14}$$

Therefore

$$\hat{\boldsymbol{\theta}} \leftarrow \operatorname*{argmax}_{\boldsymbol{\theta}} \left\{ \sum_{i=1}^{m} \log \log q(y_i; \alpha) + \sum_{i=1}^{m} \log q(\boldsymbol{x}_i \mid y_i; \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y) \right\}$$

the last item has two components, depending on the value of y

MLE: Bernoulli Distribution

Recall the definition of Bernoulli distribution, we have

$$\sum_{i=1}^{m} \log q(y_i; \alpha) = \sum_{i=1}^{m} \{\delta(y_i = +1) \log \alpha + \delta(y_i = -1) \log(1 - \alpha)\} \quad (15)$$

MLE: Bernoulli Distribution

Recall the definition of Bernoulli distribution, we have

$$\sum_{i=1}^{m} \log q(y_i;\alpha) = \sum_{i=1}^{m} \{\delta(y_i = +1)\log\alpha + \delta(y_i = -1)\log(1-\alpha)\} \tag{15}$$

Then, the value of α can be estimated from

$$\frac{d\sum_{i=1}^{m} \log q(y_i;\alpha)}{d\alpha} = \frac{\sum_{i=1}^{m}\delta(y_i = +1)}{\alpha} - \frac{\sum_{i=1}^{m}\delta(y_i = -1)}{1-\alpha} = 0 \tag{16}$$

MLE: Bernoulli Distribution

Recall the definition of Bernoulli distribution, we have

$$\sum_{i=1}^{m} \log q(y_i;\alpha) = \sum_{i=1}^{m} \{\delta(y_i = +1)\log\alpha + \delta(y_i = -1)\log(1-\alpha)\} \quad (15)$$

Then, the value of α can be estimated from

$$\frac{d\sum_{i=1}^{m} \log q(y_i;\alpha)}{d\alpha} = \frac{\sum_{i=1}^{m} \delta(y_i = +1)}{\alpha} - \frac{\sum_{i=1}^{m} \delta(y_i = -1)}{1-\alpha} = 0 \quad (16)$$

therefore,

$$\alpha = \frac{\sum_{i=1}^{m} \delta(y_i = +1)}{m} \quad (17)$$

MLE: Gaussian Distribution

The definition of multi-variate Gaussian distribution

$$q(\boldsymbol{x} \mid y; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|} \exp\Big(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\Big) \tag{18}$$

- For $y = +1$, MLE on $\boldsymbol{\mu}_+$ and $\boldsymbol{\Sigma}_+$ will only consider the samples $\boldsymbol{x}$ with $y = +1$ (assume it's S_+)

MLE: Gaussian Distribution

The definition of multi-variate Gaussian distribution

$$q(\boldsymbol{x} \mid y; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) \tag{18}$$

- For $y = +1$, MLE on $\boldsymbol{\mu}_{+}$ and $\boldsymbol{\Sigma}_{+}$ will only consider the samples $\boldsymbol{x}$ with $y = +1$ (assume it's S_{+})
- MLE on $\boldsymbol{\mu}_{+}$

$$\boldsymbol{\mu} = \frac{1}{|S_{+}|} \sum_{\boldsymbol{x}_i \in S_{+}} \boldsymbol{x}_i \tag{19}$$

MLE: Gaussian Distribution

The definition of multi-variate Gaussian distribution

$$q(\boldsymbol{x} \mid y; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|} \exp\Big(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\Big) \tag{18}$$

- For $y = +1$, MLE on $\boldsymbol{\mu}_+$ and $\boldsymbol{\Sigma}_+$ will only consider the samples $\boldsymbol{x}$ with $y = +1$ (assume it's S_+)
- MLE on $\boldsymbol{\mu}_+$

$$\boldsymbol{\mu} = \frac{1}{|S_+|}\sum_{\boldsymbol{x}_i \in S_+} \boldsymbol{x}_i \tag{19}$$

- MLE on $\boldsymbol{\Sigma}_+$

$$\boldsymbol{\Sigma}_+ = \frac{1}{|S_+|}\sum_{\boldsymbol{x}_i \in S_+} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^{\mathsf{T}} \tag{20}$$

MLE: Gaussian Distribution

The definition of multi-variate Gaussian distribution

$$q(\boldsymbol{x} \mid y; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|} \exp\Big(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\Big) \tag{18}$$

- For $y = +1$, MLE on $\boldsymbol{\mu}_+$ and $\boldsymbol{\Sigma}_+$ will only consider the samples $\boldsymbol{x}$ with $y = +1$ (assume it's S_+)
- MLE on $\boldsymbol{\mu}_+$

$$\boldsymbol{\mu} = \frac{1}{|S_+|}\sum_{\boldsymbol{x}_i \in S_+} \boldsymbol{x}_i \tag{19}$$

- MLE on $\boldsymbol{\Sigma}_+$

$$\boldsymbol{\Sigma}_+ = \frac{1}{|S_+|}\sum_{\boldsymbol{x}_i \in S_+} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^{\top} \tag{20}$$

- *Exercise*: prove equations 19 and 20 with $d = 1$

Example: Parameter Estimation

Given $N = 1000$ samples, here are the parameters

Parameter	$p(\cdot)$	$q(\cdot)$
$\boldsymbol{\mu}_+$	$[2, 0]^\mathsf{T}$	$[1.95, -0.11]^\mathsf{T}$
$\boldsymbol{\Sigma}_+$	$\begin{bmatrix} 1.0 & 0.8 \\ 0.8 & 2.0 \end{bmatrix}$	$\begin{bmatrix} 0.88 & 0.74 \\ 0.74 & 1.97 \end{bmatrix}$
$\boldsymbol{\mu}_-$	$[-2, 0]^\mathsf{T}$	$[-2.08, 0.08]^\mathsf{T}$
$\boldsymbol{\Sigma}_-$	$\begin{bmatrix} 2.0 & 0.6 \\ 0.6 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 1.88 & 0.55 \\ 0.55 & 1.07 \end{bmatrix}$

Prediction

- For a new data point $\boldsymbol{x}'$, the prediction is given as

$$q(y' \mid \boldsymbol{x}') = \frac{q(y')q(\boldsymbol{x} \mid y')}{q(\boldsymbol{x}')} \propto q(y')q(\boldsymbol{x}' \mid y') \tag{21}$$

No need to compute $q(\boldsymbol{x}')$

Prediction

- For a new data point $\boldsymbol{x}'$, the prediction is given as

$$q(y' \mid \boldsymbol{x}') = \frac{q(y')q(\boldsymbol{x} \mid y')}{q(\boldsymbol{x}')} \propto q(y')q(\boldsymbol{x}' \mid y') \tag{21}$$

No need to compute $q(\boldsymbol{x}')$

- Prediction rule

$$y' = \begin{cases} +1 & q(y' = +1 \mid \boldsymbol{x}') > q(y' = -1 \mid \boldsymbol{x}') \\ -1 & q(y' = +1 \mid \boldsymbol{x}') < q(y' = +1 \mid \boldsymbol{x}') \end{cases} \tag{22}$$

Prediction

- For a new data point $\boldsymbol{x}'$, the prediction is given as

$$q(y' \mid \boldsymbol{x}') = \frac{q(y')q(\boldsymbol{x} \mid y')}{q(\boldsymbol{x}')} \propto q(y')q(\boldsymbol{x}' \mid y') \tag{21}$$

  No need to compute $q(\boldsymbol{x}')$

- Prediction rule

$$y' = \begin{cases} +1 & q(y' = +1 \mid \boldsymbol{x}') > q(y' = -1 \mid \boldsymbol{x}') \\ -1 & q(y' = +1 \mid \boldsymbol{x}') < q(y' = +1 \mid \boldsymbol{x}') \end{cases} \tag{22}$$

- Although equation 22 looks like the one used in the Bayes optimal predictor, the prediction power is limited by

$$q(y' \mid \boldsymbol{x}') \approx p(y \mid \boldsymbol{x}) \tag{23}$$

  Again, we don't know $p(\cdot)$

Naive Bayes Classifiers

Number of Parameters

Assume $\boldsymbol{x} = (x_{\cdot,1}, \ldots, x_{\cdot,d}) \in \mathbb{R}^d$, then the number of parameters in $q(\boldsymbol{x}, y)$

- $q(y)$: 1 (α)
- $q(\boldsymbol{x} \mid y = +1)$:
  - $\boldsymbol{\mu}_+ \in \mathbb{R}^d$: d parameters
  - $\boldsymbol{\Sigma}_+ \in \mathbb{R}^{d \times d}$: d^2 parameters
- $q(\boldsymbol{x} \mid y = -1)$: $d^2 + d$ parameters

In total, we have $2d^2 + 2d + 1$ parameters

Challenge of Parameter Estimation

- When $d = 100$, we have $2d^2 + 2d + 1 = 20201$ parameters
- A close look about the covariance matrix $\mathbf{\Sigma}$ in a multivariate Gaussian distribution

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_{1,1}^2 & \cdots & \sigma_{1,d}^2 \\ \vdots & \ddots & \vdots \\ \sigma_{d,1}^2 & \cdots & \sigma_{d,d}^2 \end{bmatrix} \tag{24}$$

Challenge of Parameter Estimation

- When $d = 100$, we have $2d^2 + 2d + 1 = 20201$ parameters
- A close look about the covariance matrix $\mathbf{\Sigma}$ in a multivariate Gaussian distribution

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_{1,1}^2 & \cdots & \sigma_{1,d}^2 \\ \vdots & \ddots & \vdots \\ \sigma_{d,1}^2 & \cdots & \sigma_{d,d}^2 \end{bmatrix} \tag{24}$$

- To reduce the number of parameters, we assume

$$\sigma_{i,j} = 0 \quad \text{if } i \neq j \tag{25}$$

Diagonal Covariance Matrix

With the diagonal covariance matrix

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_{1,1}^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{d,d}^2 \end{bmatrix} \tag{26}$$

Now, the multivariate Gaussian distribution can be rewritten with

$$|\mathbf{\Sigma}| = \prod_{j=1}^{d} \sigma_{j,j}^2 \tag{27}$$

$$(\boldsymbol{x} - \boldsymbol{\mu})^\top \mathbf{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) = \sum_{j=1}^{d} \frac{(x_{\cdot,j} - \mu_j)^2}{\sigma_{j,j}^2} \tag{28}$$

Diagonal Covariance Matrix (II)

In other words

$$q(\boldsymbol{x} \mid y, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{j=1}^{d} q(x_{\cdot,j} \mid y; \mu_j, \sigma_{j,j}^2) \tag{29}$$

Diagonal Covariance Matrix (II)

In other words

$$q(\boldsymbol{x} \mid y, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{j=1}^{d} q(x_{\cdot,j} \mid y; \mu_j, \sigma_{j,j}^2) \tag{29}$$

- **Conditional Independence**: Equation 29 means, given y, each component x_j is independent of other components

Diagonal Covariance Matrix (II)

In other words

$$q(\boldsymbol{x} \mid y, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{j=1}^{d} q(x_{\cdot,j} \mid y; \mu_j, \sigma^2_{j,j}) \tag{29}$$

- **Conditional Independence**: Equation 29 means, given y, each component x_j is independent of other components
- This is a strong and naive assumption about $q(\boldsymbol{x} \mid \cdot)$

Diagonal Covariance Matrix (II)

In other words

$$q(\boldsymbol{x} \mid y, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{j=1}^{d} q(x_{\cdot,j} \mid y; \mu_j, \sigma_{j,j}^2) \quad (29)$$

- **Conditional Independence**: Equation 29 means, given y, each component x_j is independent of other components
- This is a strong and naive assumption about $q(\boldsymbol{x} \mid \cdot)$
- Together with $q(y)$, this generative model is called the **Naive Bayes** classifier

Diagonal Covariance Matrix (II)

In other words

$$q(\boldsymbol{x} \mid y, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{j=1}^{d} q(x_{\cdot,j} \mid y; \mu_j, \sigma_{j,j}^2) \tag{29}$$

- **Conditional Independence**: Equation 29 means, given y, each component x_j is independent of other components
- This is a strong and naive assumption about $q(\boldsymbol{x} \mid \cdot)$
- Together with $q(y)$, this generative model is called the **Naive Bayes** classifier
- Parameter estimation can be done per dimension

Example: Parameter Estimation

Given $N = 1000$ samples, here are the parameters

Parameter	$p(\cdot)$	$q(\cdot)$	Naive Bayes
$\boldsymbol{\mu}_+$	$[2, 0]^\top$	$[1.95, -0.11]^\top$	$[1.95, -0.11]^\top$
$\boldsymbol{\Sigma}_+$	$\begin{bmatrix} 1.0 & 0.8 \\ 0.8 & 2.0 \end{bmatrix}$	$\begin{bmatrix} 0.88 & 0.74 \\ 0.74 & 1.97 \end{bmatrix}$	$\begin{bmatrix} 0.88 & 0 \\ 0 & 1.97 \end{bmatrix}$
$\boldsymbol{\mu}_-$	$[-2, 0]^\top$	$[-2.08, 0.08]^\top$	$[-2.08, 0.08]^\top$
$\boldsymbol{\Sigma}_-$	$\begin{bmatrix} 2.0 & 0.6 \\ 0.6 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 1.88 & 0.55 \\ 0.55 & 1.07 \end{bmatrix}$	$\begin{bmatrix} 1.88 & 0 \\ 0 & 1.07 \end{bmatrix}$

Latent Variable Models

Data Generation Model, Revisited

Consider the following model again without any label information

$$p(\boldsymbol{x}) = \underbrace{\alpha \cdot \mathcal{N}(x; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)}_{c=1} + \underbrace{(1-\alpha) \cdot \mathcal{N}(x; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)}_{c=2} \tag{30}$$

Data Generation Model, Revisited

Consider the following model again without any label information

$$p(\boldsymbol{x}) = \underbrace{\alpha \cdot \mathcal{N}(x; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)}_{c=1} + \underbrace{(1-\alpha) \cdot \mathcal{N}(x; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)}_{c=2} \tag{30}$$

- No labeling information
- Instead of having two classes, now it has two components $c \in \{1, 2\}$

Data Generation Model, Revisited

Consider the following model again without any label information

$$p(\boldsymbol{x}) = \underbrace{\alpha \cdot \mathcal{N}(x; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)}_{c=1} + \underbrace{(1-\alpha) \cdot \mathcal{N}(x; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)}_{c=2} \tag{30}$$

- No labeling information
- Instead of having two classes, now it has two components $c \in \{1, 2\}$
- It is a specific case of *Gaussian mixture models*
  - A mixture model with two Gaussian components

Data Generation

The data generation process: for each data point

1. Randomly select a component c based on

$$p(c = 1) = \alpha \quad p(c = 2) = 1 - \alpha \tag{31}$$

Data Generation

The data generation process: for each data point

1. Randomly select a component c based on

$$p(c = 1) = \alpha \quad p(c = 2) = 1 - \alpha \tag{31}$$

2. Sample $\boldsymbol{x}$ from the corresponding component c

$$p(\boldsymbol{x} \mid y) = \begin{cases} \mathcal{N}(x; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) & c = 1 \\ \mathcal{N}(x; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) & c = 2 \end{cases} \tag{32}$$

Data Generation

The data generation process: for each data point

1. Randomly select a component c based on

$$p(c = 1) = \alpha \quad p(c = 2) = 1 - \alpha \tag{31}$$

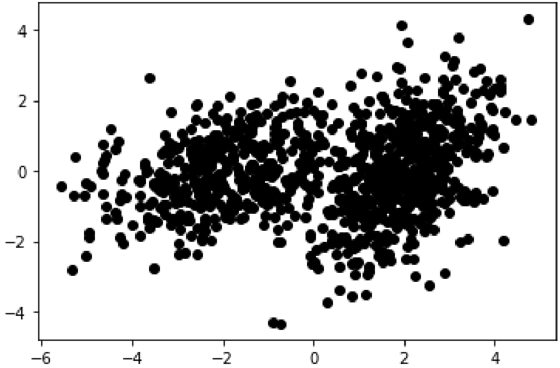
2. Sample $\boldsymbol{x}$ from the corresponding component c

$$p(\boldsymbol{x} \mid y) = \begin{cases} \mathcal{N}(x; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) & c = 1 \\ \mathcal{N}(x; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) & c = 2 \end{cases} \tag{32}$$

3. Add $\boldsymbol{x}$ to S, go to step 1

Illustration

Here is an example data set S with 1,000 samples



No label information available

The Learning Problem

Consider using the following distribution to fit the data S

$$q(\boldsymbol{x}) = \alpha \cdot \mathcal{N}(x; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1 - \alpha) \cdot \mathcal{N}(x; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \tag{33}$$

The Learning Problem

Consider using the following distribution to fit the data S

$$q(\boldsymbol{x}) = \alpha \cdot \mathcal{N}(x; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1 - \alpha) \cdot \mathcal{N}(x; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \tag{33}$$

- This is a *density estimation* problem — one of the unsupervised learning problems

The Learning Problem

Consider using the following distribution to fit the data S

$$q(\boldsymbol{x}) = \alpha \cdot \mathcal{N}(x; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1 - \alpha) \cdot \mathcal{N}(x; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \tag{33}$$

- This is a *density estimation* problem — one of the unsupervised learning problems
- The number of components in $q(\boldsymbol{x})$ is part of the assumption based on *our understanding* about the data

The Learning Problem

Consider using the following distribution to fit the data S

$$q(\boldsymbol{x}) = \alpha \cdot \mathcal{N}(x; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1 - \alpha) \cdot \mathcal{N}(x; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \tag{33}$$

- This is a *density estimation* problem — one of the unsupervised learning problems
- The number of components in $q(\boldsymbol{x})$ is part of the assumption based on *our understanding* about the data
- Without knowing the true data distribution, the number of components is treated as a hyper-parameter (predetermined before learning)

Parameter Estimation

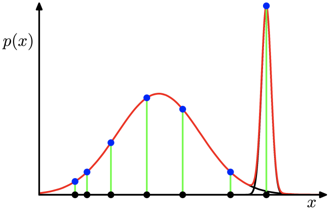
- Based on the general form of GMMs, the parameters are $\boldsymbol{\theta} = \{\alpha, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2\}$
- Given a set of training example $S = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$, the straightforward method is MLE

$$\begin{aligned} L(\boldsymbol{\theta}) &= \sum_{i=1}^{m} \log q(\boldsymbol{x}_i; \boldsymbol{\theta}) \\ &= \sum_{i=1}^{m} \log \Big(\alpha \cdot \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \\ &\quad + (1-\alpha) \cdot \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \Big) \end{aligned} \tag{34}$$

- Learning: $\boldsymbol{\theta} \leftarrow \text{argmax}_{\boldsymbol{\theta}'} L(\boldsymbol{\theta}')$

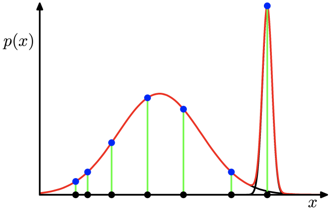
Singularity in GMM Parameter Estimation

Singularity happens when one of the mixture component only captures a single data point, which eventually leads the (log-)likelihood to ∞



Singularity in GMM Parameter Estimation

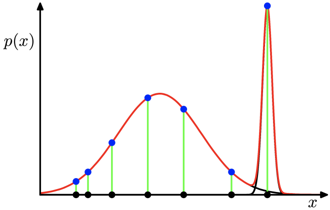
Singularity happens when one of the mixture component only captures a single data point, which eventually leads the (log-)likelihood to ∞



- It is easy to overfit the training set using GMMs, for example when $K = m$

Singularity in GMM Parameter Estimation

Singularity happens when one of the mixture component only captures a single data point, which eventually leads the (log-)likelihood to ∞



- It is easy to overfit the training set using GMMs, for example when $K = m$
- This issue does not exist when estimating parameters for a single Gaussian distribution

Gradient-based Learning

Recall the definition of $L(\boldsymbol{\theta})$

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{m} \log\left(\alpha \cdot \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1-\alpha) \cdot \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)\right) \tag{35}$$

- There is no closed form solution of $\nabla L(\boldsymbol{\theta}) = 0$
  - E.g., the value of α depends on $\{\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\}_{c=1}^{2}$, vice versa
- Gradient-based learning is still *feasible* as

$$\boldsymbol{\theta}^{(\text{new})} \leftarrow \boldsymbol{\theta}^{(\text{old})} + \eta \cdot \nabla L(\boldsymbol{\theta})$$

To rewrite equation 33 into a full probabilistic form, we introduce a random variable $z \in \{1, 2\}$, with

$$q(z = 1) = \alpha \quad q(z = 2) = 1 - \alpha \tag{36}$$

or

$$q(z) = \alpha^{\delta(z=1)}(1 - \alpha)^{\delta(z=2)} \tag{37}$$

To rewrite equation 33 into a full probabilistic form, we introduce a random variable $z \in \{1, 2\}$, with

$$q(z = 1) = \alpha \quad q(z = 2) = 1 - \alpha \tag{36}$$

or

$$q(z) = \alpha^{\delta(z=1)}(1 - \alpha)^{\delta(z=2)} \tag{37}$$

- z is a random variable and indicates the mixture component for $\boldsymbol{x}$ (a similar role as y in the classification problem)

Latent Variable Models

To rewrite equation 33 into a full probabilistic form, we introduce a random variable $z \in \{1, 2\}$, with

$$q(z = 1) = \alpha \quad q(z = 2) = 1 - \alpha \tag{36}$$

or

$$q(z) = \alpha^{\delta(z=1)}(1 - \alpha)^{\delta(z=2)} \tag{37}$$

- z is a random variable and indicates the mixture component for $\boldsymbol{x}$ (a similar role as y in the classification problem)
- z is not directly observed in the data, therefore it is a latent (random) variable.

GMM with Latent Variable

With latent variable z, we can rewrite the probabilistic model as a joint distribution over $\boldsymbol{x}$ and z

$$
\begin{aligned}
q(\boldsymbol{x}, z) &= q(z)q(\boldsymbol{x} \mid z) \\
&= \alpha^{\delta(z=1)} \cdot \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)^{\delta(z=1)} \\
&\quad \cdot (1-\alpha)^{\delta(z=2)} \cdot \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)^{\delta(z=2)} \qquad (38)
\end{aligned}
$$

GMM with Latent Variable

With latent variable z, we can rewrite the probabilistic model as a joint distribution over $\boldsymbol{x}$ and z

$$\begin{aligned} q(\boldsymbol{x}, z) &= q(z)q(\boldsymbol{x} \mid z) \\ &= \alpha^{\delta(z=1)} \cdot \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)^{\delta(z=1)} \\ &\quad \cdot (1-\alpha)^{\delta(z=2)} \cdot \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)^{\delta(z=2)} \end{aligned} \tag{38}$$

And the marginal probability $p(\boldsymbol{x})$ is the same as in equation 33

$$\begin{aligned} q(\boldsymbol{x}) &= q(z=1)q(\boldsymbol{x} \mid z=1) + q(z=2)q(\boldsymbol{x} \mid z=2) \\ &= \alpha \cdot \mathcal{N}(x; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1-\alpha) \cdot \mathcal{N}(x; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \end{aligned} \tag{39}$$

Parameter Estimation: MLE?

For each $\boldsymbol{x}_i$, we introduce a latent variable z_i as mixture component indicator, then the log likelihood is defined as

$$\begin{aligned}
\ell(\boldsymbol{\theta}) &= \sum_{i=1}^{m} \log q(\boldsymbol{x}_i, z_i) \\
&= \sum_{i=1}^{m} \log \left\{ \alpha^{\delta(z_i=1)} \cdot \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)^{\delta(z_i=1)} \right. \\
&\qquad \left. \cdot (1-\alpha)^{\delta(z_i=2)} \cdot \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)^{\delta(z_i=2)} \right\} \\
&= \sum_{i=1}^{m} \left\{ \delta(z_i = 1) \log \alpha + \delta(z_i = 1) \log \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \right. \\
&\qquad \left. \delta(z_i = 2) \log(1-\alpha) + \delta(z_i = 2) \log \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \right\}
\end{aligned} \tag{40}$$

Parameter Estimation: MLE?

For each $\boldsymbol{x}_i$, we introduce a latent variable z_i as mixture component indicator, then the log likelihood is defined as

$$\begin{aligned}
\ell(\boldsymbol{\theta}) &= \sum_{i=1}^{m} \log q(\boldsymbol{x}_i, z_i) \\
&= \sum_{i=1}^{m} \log \left\{ \alpha^{\delta(z_i=1)} \cdot \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)^{\delta(z_i=1)} \right. \\
&\quad \left. \cdot (1-\alpha)^{\delta(z_i=2)} \cdot \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)^{\delta(z_i=2)} \right\} \qquad (40) \\
&= \sum_{i=1}^{m} \left\{ \delta(z_i = 1) \log \alpha + \delta(z_i = 1) \log \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \right. \\
&\quad \left. \delta(z_i = 2) \log(1-\alpha) + \delta(z_i = 2) \log \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \right\}
\end{aligned}$$

Question: we have already know that z_i is a random variable, but $E\left[z_i = 1\right] = \alpha$?

EM Algorithm

Basic Idea

- The key challenge of GMM with latent variables is that we do not know the distributions of $\{z_i\}$

Basic Idea

- The key challenge of GMM with latent variables is that we do not know the distributions of $\{z_i\}$
- The basic idea of the EM algorithm is to alternatively address the challenge between

$$\{z_i\}_{i=1}^{m} \Leftrightarrow \boldsymbol{\theta} = \{\alpha, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2\} \tag{41}$$

Basic Idea

- The key challenge of GMM with latent variables is that we do not know the distributions of $\{z_i\}$
- The basic idea of the EM algorithm is to alternatively address the challenge between

$$\{z_i\}_{i=1}^{m} \Leftrightarrow \boldsymbol{\theta} = \{\alpha, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2\} \tag{41}$$

- Basic procedure
  1. Fix $\boldsymbol{\theta}$, estimate the distributions of $\{z_i\}_{i=1}^{m}$
  2. Fix the distribution of $\{z_i\}_{i=1}^{m}$, estimate the value of $\boldsymbol{\theta}$
  3. Go back to step 1

How to Estimate z_i?

Fix $\boldsymbol{\theta}$, we can estimate the distribution of each z_i as (with equation 38 and 39)

$$q(z_i \mid \boldsymbol{x}_i) = \frac{q(\boldsymbol{x}_i, z_i)}{q(\boldsymbol{x}_i)} \tag{42}$$

Particularly, we have

$$q(z_i = 1 \mid \boldsymbol{x}_i) = \frac{\alpha \cdot \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)}{\alpha \cdot \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1 - \alpha) \cdot \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)} \tag{43}$$

Expectation

Let γ_i be the expectation of z_i under the distribution of $q(z_i \mid \boldsymbol{x}_i)$

$$E\left[z_i\right] = \gamma_i \tag{44}$$

Expectation

Let γ_i be the expectation of z_i under the distribution of $q(z_i \mid \boldsymbol{x}_i)$

$$E[z_i] = \gamma_i \tag{44}$$

- Since z_i is a Bernoulli random variable, we also have $q(z_i = 1 \mid \boldsymbol{x}_i) = \gamma_i$

Expectation

Let γ_i be the expectation of z_i under the distribution of $q(z_i \mid \boldsymbol{x}_i)$

$$E[z_i] = \gamma_i \tag{44}$$

- Since z_i is a Bernoulli random variable, we also have $q(z_i = 1 \mid \boldsymbol{x}_i) = \gamma_i$
- Furthermore, the expectation of $\delta(z_i = 1)$ under the distribution of $q(z_i \mid \boldsymbol{x}_i)$

$$\begin{aligned} E[\delta(z_i = 1)] &= \delta(z_i = 1) \cdot q(z_i = 1 \mid \boldsymbol{x}_i) \\ &\quad + \delta(z_i = 1) \cdot q(z_i = 2 \mid \boldsymbol{x}_i) \\ &= q(z_i = 1) = \gamma_i \end{aligned} \tag{45}$$

Parameter Estimation (I)

Given

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{m} \Big\{ \delta(z_i = 1) \log \alpha + \delta(z_i = 1) \log \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$$
$$\delta(z_i = 2) \log(1 - \alpha) + \delta(z_i = 2) \log \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \Big\} \tag{46}$$

Parameter Estimation (I)

Given

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{m} \Big\{ \delta(z_i = 1) \log \alpha + \delta(z_i = 1) \log \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \tag{46}$$
$$\delta(z_i = 2) \log(1 - \alpha) + \delta(z_i = 2) \log \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \Big\}$$

To maximize $\ell(\boldsymbol{\theta})$ with respect to α we have

$$\sum_{i=1}^{m} \left\{ \frac{\delta(z_i = 1)}{\alpha} - \frac{\delta(z_i = 2)}{1 - \alpha} \right\} = 0 \tag{47}$$

Parameter Estimation (I)

Given

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{m} \Big\{\delta(z_i = 1)\log\alpha + \delta(z_i = 1)\log\mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \\ \delta(z_i = 2)\log(1-\alpha) + \delta(z_i = 2)\log\mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)\Big\} \tag{46}$$

To maximize $\ell(\boldsymbol{\theta})$ with respect to α we have

$$\sum_{i=1}^{m} \left\{\frac{\delta(z_i = 1)}{\alpha} - \frac{\delta(z_i = 2)}{1-\alpha}\right\} = 0 \tag{47}$$

and

$$\alpha \mid \boldsymbol{z} = \frac{\sum_{i=1}^{m} \delta(z_i = 1)}{\sum_{i=1}^{m}(\delta(z_i = 1) + \delta(z_i = 2))} = \frac{\sum_{i=1}^{m} \delta(z_i = 1)}{m} \tag{48}$$

which is similar to the classification example, except that z_i is a *random variable*

Parameter Estimation (II)

Without going through the details, the estimate of *mean* and *covariance* take the similar forms. For example, for the first component, we have

$$\boldsymbol{\mu}_1 \mid \boldsymbol{z} = \frac{1}{m}\sum_{i=1}^{m}\delta(z_i = 1)\boldsymbol{x}_i \tag{49}$$

$$\boldsymbol{\Sigma}_1 \mid \boldsymbol{z} = \frac{1}{m}\sum_{i=1}^{m}\delta(z_i = 1)(\boldsymbol{x}_i - \boldsymbol{\mu}_1)(\boldsymbol{x}_i - \boldsymbol{\mu}_1)^{\mathsf{T}} \tag{50}$$

Parameter Estimation (II)

Without going through the details, the estimate of *mean* and *covariance* take the similar forms. For example, for the first component, we have

$$\boldsymbol{\mu}_1 \mid \boldsymbol{z} = \frac{1}{m} \sum_{i=1}^{m} \delta(z_i = 1) \boldsymbol{x}_i \tag{49}$$

$$\boldsymbol{\Sigma}_1 \mid \boldsymbol{z} = \frac{1}{m} \sum_{i=1}^{m} \delta(z_i = 1)(\boldsymbol{x}_i - \boldsymbol{\mu}_1)(\boldsymbol{x}_i - \boldsymbol{\mu}_1)^\top \tag{50}$$

Question: how to eliminate the randomness in α, $\boldsymbol{\mu}_1$, $\boldsymbol{\Sigma}_1$ (and similarly in $\boldsymbol{\mu}_2$, $\boldsymbol{\Sigma}_2$)?

Expectation (II)

With $E[\delta(z_i = 1)] = \gamma_i$, we have

$$\begin{aligned} \alpha &= E[\alpha \mid \boldsymbol{z}] = \frac{1}{m}\sum_{i=1}^{m} E[\delta(z_i = 1)]\, \boldsymbol{x}_i \\ &= \frac{1}{m}\sum_{i=1}^{m} \gamma_i \boldsymbol{x}_i \end{aligned} \tag{51}$$

Expectation (II)

With $E[\delta(z_i = 1)] = \gamma_i$, we have

$$\begin{aligned}\alpha &= E[\alpha \mid \mathbf{z}] = \frac{1}{m}\sum_{i=1}^{m} E[\delta(z_i = 1)]\,\mathbf{x}_i \\ &= \frac{1}{m}\sum_{i=1}^{m} \gamma_i \mathbf{x}_i \end{aligned} \tag{51}$$

Similarly, we have

$$\boldsymbol{\mu}_1 = \frac{1}{m}\sum_{i=1}^{m} \gamma_i \mathbf{x}_i \qquad \boldsymbol{\mu}_2 = \frac{1}{m}\sum_{i=1}^{m} (1-\gamma_i)\mathbf{x}_i$$

$$\begin{aligned}\boldsymbol{\Sigma}_1 &= \frac{1}{m}\sum_{i=1}^{m} \gamma_i (\mathbf{x}_i - \boldsymbol{\mu}_1)(\mathbf{x}_i - \boldsymbol{\mu}_1)^{\mathsf{T}} \\ \boldsymbol{\Sigma}_2 &= \frac{1}{m}\sum_{i=1}^{m} (1-\gamma_i)(\mathbf{x}_i - \boldsymbol{\mu}_2)(\mathbf{x}_i - \boldsymbol{\mu}_2)^{\mathsf{T}} \end{aligned} \tag{52}$$

The EM Algorithm, Review

The algorithm iteratively run the following two steps:

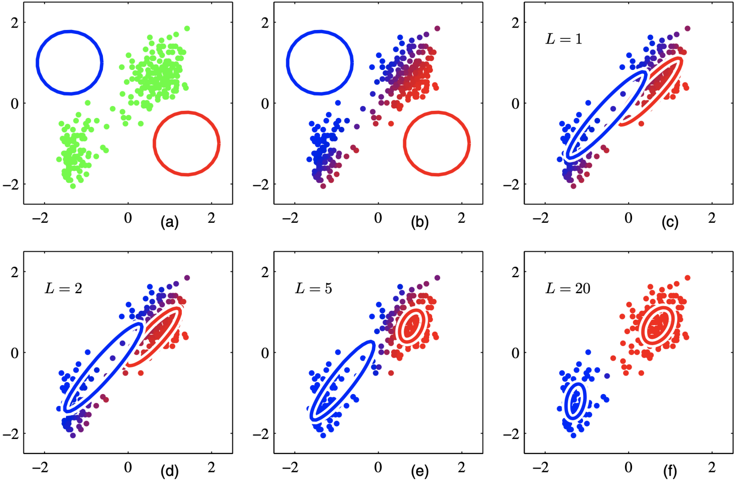
E-step Given $\boldsymbol{\theta}$, for each $\boldsymbol{x}_i$, estimate the distribution of the corresponding latent variable z_i

$$q(z_i \mid \boldsymbol{x}_i) = \frac{q(\boldsymbol{x}_i, z_i)}{q(\boldsymbol{x}_i)} \tag{53}$$

and its expectation γ_i

M-step Given $\{z_i\}_{i=1}^m$, maximize the log-likelihood function $\ell(\boldsymbol{\theta})$ and estimate the parameter $\boldsymbol{\theta}$ with $\{\gamma_i\}_{i=1}^m$

Illustration



[Bishop and Nasrabadi, 2006, Page 437]

Variational Inference (Optional)

The Computation of $q(z \mid \boldsymbol{x})$

- In the previous example, we were able to compute the analytic solution of $q(z \mid \boldsymbol{x})$ as

$$q(z \mid \boldsymbol{x}) = \frac{q(\boldsymbol{x}, z)}{q(\boldsymbol{x})} \tag{54}$$

  where $q(\boldsymbol{x}) = \sum_z q(\boldsymbol{x}, z)$

- **Challenge**: Unlike the simple case in GMMs, usually $q(\boldsymbol{x})$ is difficult to compute

$$\begin{aligned} q(\boldsymbol{x}) &= \sum_z q(\boldsymbol{x}, z) \quad \text{discrete} && (55) \\ &= \int_z q(\boldsymbol{x}, z) dz \quad \text{continuous} && (56) \end{aligned}$$

Solution

- Instead of computing $q(\boldsymbol{x})$ and then $q(z \mid \boldsymbol{x})$, we propose another distribution $q'(z \mid \boldsymbol{x})$ to approximate $q(z \mid \boldsymbol{x})$

$$q'(z \mid \boldsymbol{x}) \approx q(z \mid \boldsymbol{x}) \tag{57}$$

  where $q'(z \mid \boldsymbol{x})$ should be *simple* enough to facilitate the computation

Solution

- Instead of computing $q(\boldsymbol{x})$ and then $q(z \mid \boldsymbol{x})$, we propose another distribution $q'(z \mid \boldsymbol{x})$ to approximate $q(z \mid \boldsymbol{x})$

$$q'(z \mid \boldsymbol{x}) \approx q(z \mid \boldsymbol{x}) \tag{57}$$

  where $q'(z \mid \boldsymbol{x})$ should be *simple* enough to facilitate the computation
- The objective of finding a good approximation is the Kullback–Leibler (KL) divergence

$$\begin{aligned} \mathrm{KL}(q' \| q) &= \sum_z q'(z \mid \boldsymbol{x}) \log \frac{q'(z \mid \boldsymbol{x})}{q(z \mid \boldsymbol{x})} \quad \text{discrete} \\ &= \int_z q'(z \mid \boldsymbol{x}) \log \frac{q'(z \mid \boldsymbol{x})}{q(z \mid \boldsymbol{x})} dz \quad \text{continuous} \end{aligned}$$

KL Divergence

- $\mathrm{KL}(q' \| q) \geq 0$ and the equality holds if and only if $q' = q$

KL Divergence

- $\mathrm{KL}(q' \| q) \geq 0$ and the equality holds if and only if $q' = q$
- Consider the continuous case for the visualization purpose.

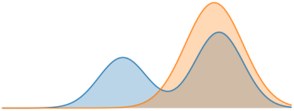
$$\mathrm{KL}(q' \| q) = \int_z q'(z \mid \boldsymbol{x}) \log \frac{q'(z \mid \boldsymbol{x})}{q(z \mid \boldsymbol{x})} dz \tag{58}$$

KL Divergence

- $\mathrm{KL}(q' \| q) \geq 0$ and the equality holds if and only if $q' = q$
- Consider the continuous case for the visualization purpose.

$$\mathrm{KL}(q' \| q) = \int_z q'(z \mid \boldsymbol{x}) \log \frac{q'(z \mid \boldsymbol{x})}{q(z \mid \boldsymbol{x})} dz \tag{58}$$

- Regardless what $q(z \mid \boldsymbol{x})$ looks like, we decide to define $q'(z \mid \boldsymbol{x})$ for simplicity

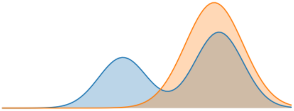


KL Divergence

- $\mathrm{KL}(q' \| q) \geq 0$ and the equality holds if and only if $q' = q$
- Consider the continuous case for the visualization purpose.

$$\mathrm{KL}(q' \| q) = \int_z q'(z \mid \boldsymbol{x}) \log \frac{q'(z \mid \boldsymbol{x})}{q(z \mid \boldsymbol{x})} dz \tag{58}$$

- Regardless what $q(z \mid \boldsymbol{x})$ looks like, we decide to define $q'(z \mid \boldsymbol{x})$ for simplicity



- Because of $q(z \mid \boldsymbol{x})$ in equation 58, the challenge still exists

ELBo

The learning objective for $q'(z \mid \boldsymbol{x})$ is

$$\mathrm{KL}(q' \| q) \quad = \quad \int_z q'(z \mid \boldsymbol{x}) \log \frac{q'(z \mid \boldsymbol{x})}{q(z \mid \boldsymbol{x})} dz$$

ELBo

The learning objective for $q'(z \mid \boldsymbol{x})$ is

$$
\begin{aligned}
\mathrm{KL}(q' \| q) &= \int_z q'(z \mid \boldsymbol{x}) \log \frac{q'(z \mid \boldsymbol{x})}{q(z \mid \boldsymbol{x})} dz \\
&= \int_z q'(z \mid \boldsymbol{x}) \log \frac{q'(z \mid \boldsymbol{x}) q(\boldsymbol{x})}{q(z, \boldsymbol{x})} dz \\
&= \int_z q'(z \mid \boldsymbol{x}) \log \frac{q'(z \mid \boldsymbol{x}) q(\boldsymbol{x})}{q(\boldsymbol{x} \mid z) q(z)} dz
\end{aligned}
$$

ELBo

The learning objective for $q'(z \mid \boldsymbol{x})$ is

$$
\begin{aligned}
\mathrm{KL}(q' \| q) &= \int_z q'(z \mid \boldsymbol{x}) \log \frac{q'(z \mid \boldsymbol{x})}{q(z \mid \boldsymbol{x})} dz \\
&= \int_z q'(z \mid \boldsymbol{x}) \log \frac{q'(z \mid \boldsymbol{x}) q(\boldsymbol{x})}{q(z, \boldsymbol{x})} dz \\
&= \int_z q'(z \mid \boldsymbol{x}) \log \frac{q'(z \mid \boldsymbol{x}) q(\boldsymbol{x})}{q(\boldsymbol{x} \mid z) q(z)} dz \\
&= \int_z q'(z \mid \boldsymbol{x}) \Big\{ -\log q(\boldsymbol{x} \mid z) + \log \frac{q'(z \mid \boldsymbol{x})}{q(z)} + \log q(\boldsymbol{x}) \Big\} dz \\
&= -E \left[\log q(\boldsymbol{x} \mid z) \right] + \mathrm{KL}(q'(z \mid \boldsymbol{x}) \| q(z)) + \log q(\boldsymbol{x})
\end{aligned}
$$

ELBo

The learning objective for $q'(z \mid \boldsymbol{x})$ is

$$
\begin{aligned}
\mathrm{KL}(q' \| q) &= \int_z q'(z \mid \boldsymbol{x}) \log \frac{q'(z \mid \boldsymbol{x})}{q(z \mid \boldsymbol{x})} dz \\
&= \int_z q'(z \mid \boldsymbol{x}) \log \frac{q'(z \mid \boldsymbol{x}) q(\boldsymbol{x})}{q(z, \boldsymbol{x})} dz \\
&= \int_z q'(z \mid \boldsymbol{x}) \log \frac{q'(z \mid \boldsymbol{x}) q(\boldsymbol{x})}{q(\boldsymbol{x} \mid z) q(z)} dz \\
&= \int_z q'(z \mid \boldsymbol{x}) \Big\{ -\log q(\boldsymbol{x} \mid z) + \log \frac{q'(z \mid \boldsymbol{x})}{q(z)} + \log q(\boldsymbol{x}) \Big\} dz \\
&= -E\left[\log q(\boldsymbol{x} \mid z)\right] + \mathrm{KL}(q'(z \mid \boldsymbol{x}) \| q(z)) + \log q(\boldsymbol{x}) \\
&= -\mathrm{ELBo} + \log q(\boldsymbol{x})
\end{aligned}
$$

Minimize $\mathrm{KL}(q' \| q)$ is equivalent to maximize the Evidence Lower Bound (ELBo)

Reference

Bishop, C. M. and Nasrabadi, N. M. (2006).
Pattern recognition and machine learning, volume 4.
Springer.