

# CS 6316 Machine Learning

## Generative Models

---

Yangfeng Ji

Department of Computer Science  
University of Virginia



ENGINEERING

## 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  $x \in \mathcal{X}$  according to  $\mathcal{D}$
3. annotate it using the labeling function  $f$  as  $y = f(x)$

[From Lecture 02]

# Example

Here is an data generation model

$$p(x) = \underbrace{0.6 \cdot \mathcal{N}(x; \mu_+, \Sigma_+)}_{y=+1} + \underbrace{0.4 \cdot \mathcal{N}(x; \mu_-, \Sigma_-)}_{y=-1} \quad (1)$$

with

- ▶  $\mu_+ = [2, 0]^T$
- ▶  $\Sigma_+ = \begin{bmatrix} 1.0 & 0.8 \\ 0.8 & 2.0 \end{bmatrix}$
- ▶  $\mu_- = [-2, 0]^T$
- ▶  $\Sigma_- = \begin{bmatrix} 2.0 & 0.6 \\ 0.6 & 1.0 \end{bmatrix}$

## Example (II)

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

$$p(y = +1) = 0.6 \quad (2)$$

$$p(y = -1) = 1 - p(y = +1) = 0.4 \quad (3)$$

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

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

# Data Generation

The specific data generation process:  
for each data point

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

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

# Data Generation

The specific data generation process:  
for each data point

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

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

2. Sample  $x$  from the corresponding component based on the value of  $y$

$$p(x | y) = \begin{cases} \mathcal{N}(x; \mu_+, \Sigma_+) & y = +1 \\ \mathcal{N}(x; \mu_-, \Sigma_-) & y = -1 \end{cases} \quad (7)$$

# Data Generation

The specific data generation process:  
for each data point

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

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

2. Sample  $x$  from the corresponding component based on the value of  $y$

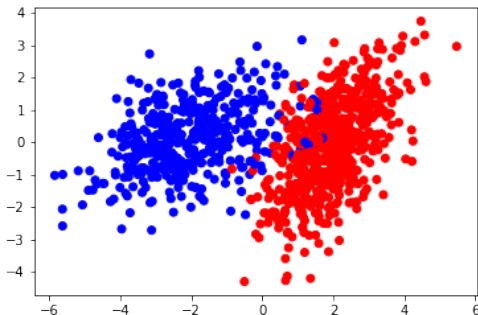
$$p(x | y) = \begin{cases} \mathcal{N}(x; \mu_+, \Sigma_+) & y = +1 \\ \mathcal{N}(x; \mu_-, \Sigma_-) & y = -1 \end{cases} \quad (7)$$

3. Add  $(x, y)$  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

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

where  $w$  is the model parameter

# Discriminative Models for Classification

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

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

where  $w$  is the model parameter

- ▶ Other examples
  - ▶ AdaBoost (lecture 05)
  - ▶ SVMs (lecture 07)
  - ▶ Feed-forward neural network (lecture 08)

# 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  $y \in \{-1, +1\}$
  - ▶  $p(x \mid y = +1)$  where  $x \in \mathbb{R}^d$
  - ▶  $p(x \mid y = -1)$  where  $x \in \mathbb{R}^d$

# 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  $y \in \{-1, +1\}$
  - ▶  $p(x \mid y = +1)$  where  $x \in \mathbb{R}^d$
  - ▶  $p(x \mid y = -1)$  where  $x \in \mathbb{R}^d$
- ▶ 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
$p(y)$	$q(y)$
$p(x \mid y = +1)$	$q(x \mid y = +1)$
$p(x \mid y = -1)$	$q(x \mid y = -1)$

# 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 data*

# Defining Distributions

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

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

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

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 data*

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

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

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

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

$$p(x \mid y = +1) = \mathcal{N}(x; \mu_+, \Sigma_+) \quad (10)$$

where  $\mu_+$  and  $\Sigma_+$  are the parameters

# Defining Distributions

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

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

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

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

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

$$p(x \mid y = +1) = \mathcal{N}(x; \mu_+, \Sigma_+) \quad (10)$$

where  $\mu_+$  and  $\Sigma_+$  are the parameters

- ▶ Similarly, for  $p(x \mid y = -1)$

$$p(x \mid y = -1) = \mathcal{N}(x; \mu_-, \Sigma_-) \quad (11)$$

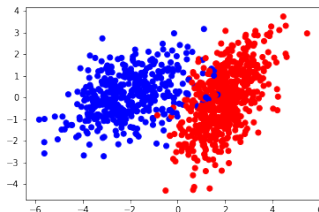
where  $\mu_-$  and  $\Sigma_-$  are the parameters

# Parameter Estimation

- ▶ The collection of the parameters

$$\theta = \{\alpha, \mu_+, \Sigma_+, \mu_-, \Sigma_-\} \quad (12)$$

- ▶ Training data  $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$

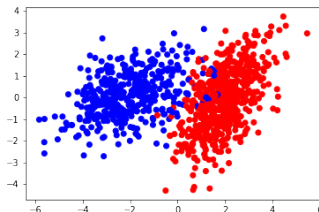


# Parameter Estimation

- ▶ The collection of the parameters

$$\theta = \{\alpha, \mu_+, \Sigma_+, \mu_-, \Sigma_-\} \quad (12)$$

- ▶ Training data  $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$



- ▶ Learning algorithm: Maximum Likelihood Estimation (MLE)

# Maximum Likelihood Estimation (MLE)

MLE defined on the whole distribution  $q(x, y)$

$$\theta \leftarrow \operatorname{argmax}_{\theta'} \sum_{i=1}^m \log q(x_i, y_i; \theta') \quad (13)$$

# Maximum Likelihood Estimation (MLE)

MLE defined on the whole distribution  $q(x, y)$

$$\theta \leftarrow \operatorname{argmax}_{\theta'} \sum_{i=1}^m \log q(x_i, y_i; \theta') \quad (13)$$

Based on the chain rule of probability

$$q(x, y; \theta) = q(y; \alpha)q(x \mid y; \mu_y, \Sigma_y), \quad (14)$$



# Maximum Likelihood Estimation (MLE)

MLE defined on the whole distribution  $q(x, y)$

$$\theta \leftarrow \operatorname{argmax}_{\theta'} \sum_{i=1}^m \log q(x_i, y_i; \theta') \quad (13)$$

Based on the chain rule of probability

$$q(x, y; \theta) = q(y; \alpha)q(x \mid y; \mu_y, \Sigma_y), \quad (14)$$

Therefore

$$\hat{\theta} \leftarrow \operatorname{argmax}_{\theta} \left\{ \sum_{i=1}^m \log \log q(y_i; \alpha) + \sum_{i=1}^m \log q(x_i \mid y_i; \mu_y, \Sigma_y) \right\}$$

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

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

Then, the value of  $\alpha$  can be estimated from

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

# 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  $\alpha$  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

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

- For  $y = +1$ , MLE on  $\mu_+$  and  $\Sigma_+$  will only consider the samples  $x$  with  $y = +1$  (assume it's  $S_+$ )

# MLE: Gaussian Distribution

The definition of multi-variate Gaussian distribution

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

- ▶ For  $y = +1$ , MLE on  $\mu_+$  and  $\Sigma_+$  will only consider the samples  $x$  with  $y = +1$  (assume it's  $S_+$ )
- ▶ MLE on  $\mu_+$

$$\mu = \frac{1}{|S_+|} \sum_{x_i \in S_+} x_i \quad (19)$$

# MLE: Gaussian Distribution

The definition of multi-variate Gaussian distribution

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

- ▶ For  $y = +1$ , MLE on  $\mu_+$  and  $\Sigma_+$  will only consider the samples  $x$  with  $y = +1$  (assume it's  $S_+$ )
- ▶ MLE on  $\mu_+$

$$\mu = \frac{1}{|S_+|} \sum_{x_i \in S_+} x_i \quad (19)$$

- ▶ MLE on  $\Sigma_+$

$$\Sigma_+ = \sum_{x_i \in S_+} (x_i - \mu)(x_i - \mu)^\top \quad (20)$$

# MLE: Gaussian Distribution

The definition of multi-variate Gaussian distribution

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

- ▶ For  $y = +1$ , MLE on  $\mu_+$  and  $\Sigma_+$  will only consider the samples  $x$  with  $y = +1$  (assume it's  $S_+$ )
- ▶ MLE on  $\mu_+$

$$\mu = \frac{1}{|S_+|} \sum_{x_i \in S_+} x_i \quad (19)$$

- ▶ MLE on  $\Sigma_+$

$$\Sigma_+ = \sum_{x_i \in S_+} (x_i - \mu)(x_i - \mu)^\top \quad (20)$$

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



# Example: Parameter Estimation

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

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

# Prediction

- For a new data point  $x'$ , the prediction is given as

$$q(y' | x') = \frac{q(y')q(x | y')}{q(x')} \propto q(y')q(x' | y') \quad (21)$$

No need to compute  $q(x')$

# Prediction

- ▶ For a new data point  $x'$ , the prediction is given as

$$q(y' | x') = \frac{q(y')q(x | y')}{q(x')} \propto q(y')q(x' | y') \quad (21)$$

No need to compute  $q(x')$

- ▶ Prediction rule

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

# Prediction

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

$$q(y' | \mathbf{x}') = \frac{q(y')q(\mathbf{x} | y')}{q(\mathbf{x}')} \propto q(y')q(\mathbf{x}' | y') \quad (21)$$

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

- ▶ Prediction rule

$$y' = \begin{cases} +1 & q(y' = +1 | \mathbf{x}') > q(y' = -1 | \mathbf{x}') \\ -1 & q(y' = +1 | \mathbf{x}') < q(y' = +1 | \mathbf{x}') \end{cases} \quad (22)$$

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

$$q(y' | \mathbf{x}') \approx p(y | \mathbf{x}) \quad (23)$$

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

# Naive Bayes Classifiers

---

# Number of Parameters

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

- ▶  $q(y)$ : 1 ( $\alpha$ )
- ▶  $q(\mathbf{x} \mid y = +1)$ :
  - ▶  $\mu_+ \in \mathbb{R}^d$ :  $d$  parameters
  - ▶  $\Sigma_+ \in \mathbb{R}^{d \times d}$ :  $d^2$  parameters
- ▶  $q(\mathbf{x} \mid y = -1)$ :  $d^2 + d$  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  $\Sigma$  in a multivariate Gaussian distribution

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

# Challenge of Parameter Estimation

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

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

- ▶ To reduce the number of parameters, we assume

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



# Diagonal Covariance Matrix

With the diagonal covariance matrix

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

Now, the multivariate Gaussian distribution can be rewritten with

- ▶  $|\Sigma| = \prod_{j=1}^d \sigma_{j,j}^2$
- ▶ assume  $\mu = 0$  for simplicity

$$(x - \mu)^\top \Sigma^{-1} (x - \mu) = \sum_{j=1}^d \frac{(x_{\cdot,j} - \mu_j)^2}{\sigma_{j,j}^2} \quad (27)$$

# Diagonal Covariance Matrix (II)

In other words

$$q(\mathbf{x} \mid y, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{j=1}^d q(x_{\cdot,j} \mid y; \mu_j, \sigma_{j,j}^2) \quad (28)$$

# Diagonal Covariance Matrix (II)

In other words

$$q(\mathbf{x} \mid y, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{j=1}^d q(x_{\cdot,j} \mid y; \mu_j, \sigma_{j,j}^2) \quad (28)$$

- **Conditional Independence:** Equation 28 means, given  $y$ , each component  $x_j$  is independent of other components

# Diagonal Covariance Matrix (II)

In other words

$$q(\mathbf{x} \mid y, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{j=1}^d q(x_{\cdot,j} \mid y; \mu_j, \sigma_{j,j}^2) \quad (28)$$

- ▶ **Conditional Independence:** Equation 28 means, given  $y$ , each component  $x_j$  is independent of other components
- ▶ This is a strong and **naive** assumption about  $q(\mathbf{x} \mid \cdot)$

# Diagonal Covariance Matrix (II)

In other words

$$q(\mathbf{x} \mid y, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{j=1}^d q(x_{\cdot,j} \mid y; \mu_j, \sigma_{j,j}^2) \quad (28)$$

- ▶ **Conditional Independence:** Equation 28 means, given  $y$ , each component  $x_j$  is independent of other components
- ▶ This is a strong and **naive** assumption about  $q(\mathbf{x} \mid \cdot)$
- ▶ Together with  $q(y)$ , this generative model is called the **Naive Bayes** classifier

# Diagonal Covariance Matrix (II)

In other words

$$q(\mathbf{x} \mid y, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{j=1}^d q(x_{\cdot,j} \mid y; \mu_j, \sigma_{j,j}^2) \quad (28)$$

- ▶ **Conditional Independence:** Equation 28 means, given  $y$ , each component  $x_j$  is independent of other components
- ▶ This is a strong and **naive** assumption about  $q(\mathbf{x} \mid \cdot)$
- ▶ 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
$\mu_+$	$[2, 0]^\top$	$[1.95, -0.11]^\top$	$[1.95, -0.11]^\top$
$\Sigma_+$	$\begin{bmatrix} 1.0 & 0.8 \\ 0.8 & 2.0 \end{bmatrix}$	$\begin{bmatrix} 0.88 & 0.74 \\ 0.74 & 1.97 \end{bmatrix}$	$\begin{bmatrix} 0.88 & 0 \\ 0 & 1.97 \end{bmatrix}$
$\mu_-$	$[-2, 0]^\top$	$[-2.08, 0.08]^\top$	$[-2.08, 0.08]^\top$
$\Sigma_-$	$\begin{bmatrix} 2.0 & 0.6 \\ 0.6 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 1.88 & 0.55 \\ 0.55 & 1.07 \end{bmatrix}$	$\begin{bmatrix} 1.88 & 0 \\ 0 & 1.07 \end{bmatrix}$

# Latent Variable Models

---



# Data Generation Model, Revisited

Consider the following model again without any label information

$$p(x) = \underbrace{\alpha \cdot \mathcal{N}(x; \mu_1, \Sigma_1)}_{c=1} + \underbrace{(1 - \alpha) \cdot \mathcal{N}(x; \mu_2, \Sigma_2)}_{c=2} \quad (29)$$

# Data Generation Model, Revisited

Consider the following model again without any label information

$$p(x) = \underbrace{\alpha \cdot \mathcal{N}(x; \mu_1, \Sigma_1)}_{c=1} + \underbrace{(1 - \alpha) \cdot \mathcal{N}(x; \mu_2, \Sigma_2)}_{c=2} \quad (29)$$

- ▶ 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

$$p(x) = \underbrace{\alpha \cdot \mathcal{N}(x; \mu_1, \Sigma_1)}_{c=1} + \underbrace{(1 - \alpha) \cdot \mathcal{N}(x; \mu_2, \Sigma_2)}_{c=2} \quad (29)$$

- ▶ 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

$$p(c = 1) = \alpha \quad p(c = 2) = 1 - \alpha \quad (30)$$

# Data Generation

The data generation process: for each data point

1. Randomly select a component  $c$  based on

$$p(c = 1) = \alpha \quad p(c = 2) = 1 - \alpha \quad (30)$$

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

$$p(\mathbf{x} \mid y) = \begin{cases} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) & c = 1 \\ \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) & c = 2 \end{cases} \quad (31)$$

# Data Generation

The data generation process: for each data point

1. Randomly select a component  $c$  based on

$$p(c = 1) = \alpha \quad p(c = 2) = 1 - \alpha \quad (30)$$

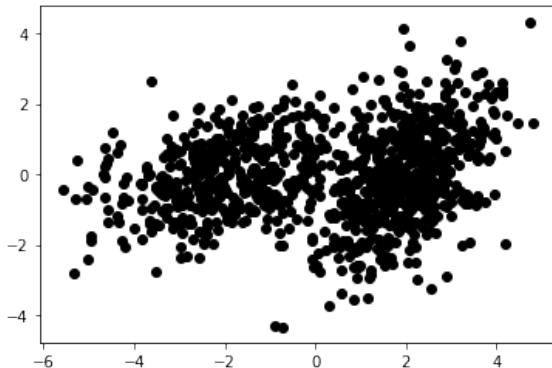
2. Sample  $x$  from the corresponding component  $c$

$$p(x \mid y) = \begin{cases} \mathcal{N}(x; \mu_1, \Sigma_1) & c = 1 \\ \mathcal{N}(x; \mu_2, \Sigma_2) & c = 2 \end{cases} \quad (31)$$

3. Add  $x$  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  $S$

$$q(x) = \alpha \cdot \mathcal{N}(x; \mu_1, \Sigma_1) + (1 - \alpha) \cdot \mathcal{N}(x; \mu_2, \Sigma_2) \quad (32)$$



# The Learning Problem

Consider using the following distribution to fit the data  $S$

$$q(x) = \alpha \cdot \mathcal{N}(x; \mu_1, \Sigma_1) + (1 - \alpha) \cdot \mathcal{N}(x; \mu_2, \Sigma_2) \quad (32)$$

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

# The Learning Problem

Consider using the following distribution to fit the data  $S$

$$q(\mathbf{x}) = \alpha \cdot \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1 - \alpha) \cdot \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \quad (32)$$

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

# The Learning Problem

Consider using the following distribution to fit the data  $S$

$$q(x) = \alpha \cdot \mathcal{N}(x; \mu_1, \Sigma_1) + (1 - \alpha) \cdot \mathcal{N}(x; \mu_2, \Sigma_2) \quad (32)$$

- ▶ This is a *density estimation* problem — one of the unsupervised learning problems
- ▶ The number of components in  $q(x)$  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

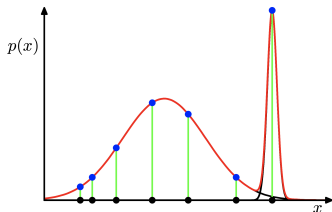
- ▶ Based on the general form of GMMs, the parameters are  $\theta = \{\alpha, \mu_1, \Sigma_1, \mu_2, \Sigma_2\}$
- ▶ Given a set of training example  $S = \{x_1, \dots, x_m\}$ , the straightforward method is MLE

$$\begin{aligned} L(\theta) &= \sum_{i=1}^m \log q(x_i; \theta) \\ &= \sum_{i=1}^m \log \left( \alpha \cdot \mathcal{N}(x_i; \mu_1, \Sigma_1) \right. \\ &\quad \left. + (1 - \alpha) \cdot \mathcal{N}(x_i; \mu_2, \Sigma_2) \right) \end{aligned} \quad (33)$$

- ▶ Learning:  $\theta \leftarrow \operatorname{argmax}_{\theta'} L(\theta')$

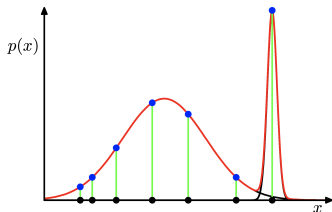
# Singularity in GMM Parameter Estimation

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



# Singularity in GMM Parameter Estimation

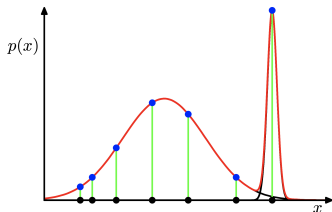
Singularity happens when one of the mixture component only captures a single data point, which eventually leads the (log-)likelihood to  $\infty$



- 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  $\infty$



- ▶ 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  $L(\boldsymbol{\theta})$

$$L(\boldsymbol{\theta}) = \sum_{i=1}^m \log \left( \alpha \cdot \mathcal{N}(x_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1 - \alpha) \cdot \mathcal{N}(x_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \right) \quad (34)$$

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

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



# Latent Variable Models

To rewrite equation 32 into a full probabilistic form, we introduce a random variable  $z \in \{1, 2\}$ , with

$$q(z = 1) = \alpha \quad q(z = 2) = 1 - \alpha \quad (35)$$

or

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

# Latent Variable Models

To rewrite equation 32 into a full probabilistic form, we introduce a random variable  $z \in \{1, 2\}$ , with

$$q(z = 1) = \alpha \quad q(z = 2) = 1 - \alpha \quad (35)$$

or

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

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

# Latent Variable Models

To rewrite equation 32 into a full probabilistic form, we introduce a random variable  $z \in \{1, 2\}$ , with

$$q(z = 1) = \alpha \quad q(z = 2) = 1 - \alpha \quad (35)$$

or

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

- ▶  $z$  is a random variable and indicates the mixture component for  $x$  (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  $x$  and  $z$

$$\begin{aligned} q(x, z) &= q(z)q(x | z) \\ &= \alpha^{\delta(z=1)} \cdot \mathcal{N}(x; \mu_1, \Sigma_1)^{\delta(z=1)} \\ &\quad \cdot (1 - \alpha)^{\delta(z=2)} \cdot \mathcal{N}(x; \mu_2, \Sigma_2)^{\delta(z=2)} \end{aligned} \quad (37)$$

# GMM with Latent Variable

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

$$\begin{aligned}q(x, z) &= q(z)q(x | z) \\&= \alpha^{\delta(z=1)} \cdot \mathcal{N}(x; \mu_1, \Sigma_1)^{\delta(z=1)} \\&\quad \cdot (1 - \alpha)^{\delta(z=2)} \cdot \mathcal{N}(x; \mu_2, \Sigma_2)^{\delta(z=2)}\end{aligned}\quad (37)$$

And the marginal probability  $p(x)$  is the same as in equation 32

$$\begin{aligned}q(x) &= q(z = 1)q(x | z = 1) + q(z = 2)q(x | z = 2) \\&= \alpha \cdot \mathcal{N}(x; \mu_1, \Sigma_1) + (1 - \alpha) \cdot \mathcal{N}(x; \mu_2, \Sigma_2)\end{aligned}\quad (38)$$

# Parameter Estimation: MLE?

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

$$\begin{aligned}\ell(\boldsymbol{\theta}) &= \sum_{i=1}^m \log q(\mathbf{x}_i, z_i) \\ &= \sum_{i=1}^m \log \left\{ \alpha^{\delta(z_i=1)} \cdot \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)^{\delta(z_i=1)} \right. \\ &\quad \left. \cdot (1 - \alpha)^{\delta(z_i=2)} \cdot \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)^{\delta(z_i=2)} \right\} \\ &= \sum_{i=1}^m \left\{ \delta(z_i = 1) \log \alpha + \delta(z_i = 1) \log \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \right. \\ &\quad \left. \delta(z_i = 2) \log(1 - \alpha) + \delta(z_i = 2) \log \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \right\}\end{aligned}\tag{39}$$

# Parameter Estimation: MLE?

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

$$\begin{aligned}\ell(\boldsymbol{\theta}) &= \sum_{i=1}^m \log q(\mathbf{x}_i, z_i) \\ &= \sum_{i=1}^m \log \left\{ \alpha^{\delta(z_i=1)} \cdot \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)^{\delta(z_i=1)} \right. \\ &\quad \left. \cdot (1 - \alpha)^{\delta(z_i=2)} \cdot \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)^{\delta(z_i=2)} \right\} \\ &= \sum_{i=1}^m \left\{ \delta(z_i = 1) \log \alpha + \delta(z_i = 1) \log \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \right. \\ &\quad \left. \delta(z_i = 2) \log(1 - \alpha) + \delta(z_i = 2) \log \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \right\}\end{aligned}\tag{39}$$

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

# 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

$$\{z_i\}_{i=1}^m \Leftrightarrow \theta = \{\alpha, \mu_1, \Sigma_1, \mu_2, \Sigma_2\} \quad (40)$$

# 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

$$\{z_i\}_{i=1}^m \Leftrightarrow \theta = \{\alpha, \mu_1, \Sigma_1, \mu_2, \Sigma_2\} \quad (40)$$

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

# How to Estimate $z_i$ ?

Fix  $\theta$ , we can estimate the distribution of each  $z_i$  as (with equation 37 and 38)

$$q(z_i \mid \mathbf{x}_i) = \frac{q(\mathbf{x}_i, z_i)}{q(\mathbf{x}_i)} \quad (41)$$

Particularly, we have

$$q(z_i = 1 \mid \mathbf{x}_i) = \frac{\alpha \cdot \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)}{\alpha \cdot \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + (1 - \alpha) \cdot \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)} \quad (42)$$

# Expectation

Let  $\gamma_i$  be the **expectation** of  $z_i$

$$E[z_i] = \gamma_i \quad (43)$$

# Expectation

Let  $\gamma_i$  be the **expectation** of  $z_i$

$$E[z_i] = \gamma_i \quad (43)$$

- ▶ Since  $z_i$  is a Bernoulli random variable, we also have  
 $p(z_i = 1) = \gamma_i$

# Expectation

Let  $\gamma_i$  be the **expectation** of  $z_i$

$$E[z_i] = \gamma_i \quad (43)$$

- ▶ Since  $z_i$  is a Bernoulli random variable, we also have

$$p(z_i = 1) = \gamma_i$$

- ▶ Furthermore, the expectation of  $\delta(z_i = 1)$

$$\begin{aligned} E[\delta(z_i = 1)] &= \delta(z_i = 1) \cdot p(z_i = 1) + \delta(z_i = 1) \cdot p(z_i = 2) \\ &= p(z_i = 1) = \gamma_i \end{aligned} \quad (44)$$

# Parameter Estimation (I)

Given

$$\begin{aligned} \ell(\boldsymbol{\theta}) = \sum_{i=1}^m \{ & \delta(z_i = 1) \log \alpha + \delta(z_i = 1) \log \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \\ & \delta(z_i = 2) \log(1 - \alpha) + \delta(z_i = 2) \log \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \} \end{aligned} \quad (45)$$



# Parameter Estimation (I)

Given

$$\begin{aligned} \ell(\boldsymbol{\theta}) = \sum_{i=1}^m \{ & \delta(z_i = 1) \log \alpha + \delta(z_i = 1) \log \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \\ & \delta(z_i = 2) \log(1 - \alpha) + \delta(z_i = 2) \log \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \} \end{aligned} \quad (45)$$

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

$$\sum_{i=1}^m \left\{ \frac{\delta(z_i = 1)}{\alpha} - \frac{\delta(z_i = 2)}{1 - \alpha} \right\} = 0 \quad (46)$$

# Parameter Estimation (I)

Given

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^m \left\{ \delta(z_i = 1) \log \alpha + \delta(z_i = 1) \log \mathcal{N}(x_i; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \right. \\ \left. \delta(z_i = 2) \log(1 - \alpha) + \delta(z_i = 2) \log \mathcal{N}(x_i; \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \right\} \quad (45)$$

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

$$\sum_{i=1}^m \left\{ \frac{\delta(z_i = 1)}{\alpha} - \frac{\delta(z_i = 2)}{1 - \alpha} \right\} = 0 \quad (46)$$

and

$$\alpha \mid \mathbf{z} = \frac{\sum_{i=1}^m \delta(z_i = 1)}{\sum_{i=1}^m (\delta(z_i = 1) + \delta(z_i = 2))} = \frac{\sum_{i=1}^m \delta(z_i = 1)}{m} \quad (47)$$

which is similar to the classification example, except  $z_i$  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

$$\mu_1 | z = \frac{1}{m} \sum_{i=1}^m \delta(z_i = 1) x_i \quad (48)$$

$$\Sigma_1 | z = \frac{1}{m} \sum_{i=1}^m \delta(z_i = 1) (x_i - \mu_1)(x_i - \mu_1)^\top \quad (49)$$

# 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

$$\mu_1 | z = \frac{1}{m} \sum_{i=1}^m \delta(z_i = 1) x_i \quad (48)$$

$$\Sigma_1 | z = \frac{1}{m} \sum_{i=1}^m \delta(z_i = 1) (x_i - \mu_1)(x_i - \mu_1)^T \quad (49)$$

*Question:* how to eliminate the randomness in  $\alpha, \mu_1, \Sigma_1$  (and similarly in  $\mu_2, \Sigma_2$ )?

## Expectation (II)

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

$$\begin{aligned}\alpha &= E [\alpha \mid \mathbf{z}] = \sum_{i=1}^m \frac{1}{m} E [\delta(z_i = 1)] \mathbf{x}_i \\ &= \sum_{i=1}^m \gamma_i \mathbf{x}_i\end{aligned}\tag{50}$$

# Expectation (II)

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

$$\begin{aligned}\alpha &= E[\alpha | \mathbf{z}] = \sum_{i=1}^m \frac{1}{m} E[\delta(z_i = 1)] \mathbf{x}_i \\ &= \sum_{i=1}^m \gamma_i \mathbf{x}_i\end{aligned}\tag{50}$$

Similarly, we have

$$\begin{aligned}\mu_1 &= \frac{1}{m} \sum_{i=1}^m \gamma_i \mathbf{x}_i & \mu_2 &= \frac{1}{m} \sum_{i=1}^m (1 - \gamma_i) \mathbf{x}_i \\ \Sigma_1 &= \frac{1}{m} \sum_{i=1}^m \gamma_i (\mathbf{x}_i - \mu_1)(\mathbf{x}_i - \mu_1)^\top \\ \Sigma_2 &= \frac{1}{m} \sum_{i=1}^m (1 - \gamma_i) (\mathbf{x}_i - \mu_2)(\mathbf{x}_i - \mu_2)^\top\end{aligned}\tag{51}$$

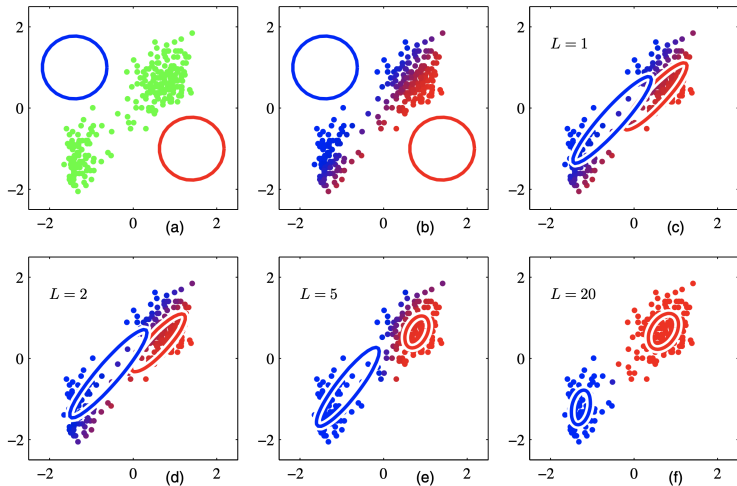
# The EM Algorithm, review

The algorithm iteratively run the following two steps:

**E-step** Given  $\theta$ , for each  $x_i$ , estimate the distribution of the corresponding latent variable  $z_i$  and its **expectation**  $\gamma_i$

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

# Illustration



[Bishop, 2006, Page 437]



# Reference



Bishop, C. M. (2006).  
*Pattern recognition and machine learning*.  
springer.