CS 6316 Machine Learning

Generative Models

Yangfeng Ji

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 02]

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

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

Example (II)

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

$$
\begin{aligned}
p(y=+1) &= 0.6 && (2)\\
p(y=-1) &= 1-p(y=+1)=0.4 && (3)\\
p(\boldsymbol{x} \mid y=+1) &= \mathcal{N}(x;\boldsymbol{\mu}_{+},\boldsymbol{\Sigma}_{+}) && (4)\\
p(\boldsymbol{x} \mid y=-1) &= \mathcal{N}(x;\boldsymbol{\mu}_{-},\boldsymbol{\Sigma}_{-}) && (5)
\end{aligned}
$$

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

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

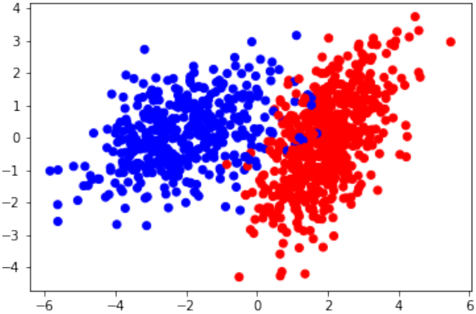
2. Sample $\boldsymbol{x}$ from the corresponding component based on the value of y

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

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
  - 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 $y \in \{-1, +1\}$
  - $p(\boldsymbol{x} \mid y = +1)$ 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 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

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

  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 data*

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

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

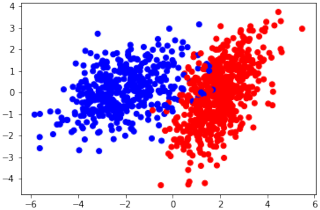
  where $\boldsymbol{\mu}_-$ and $\boldsymbol{\Sigma}_-$ are the parameters

Parameter Estimation

- The collection of the parameters

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

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

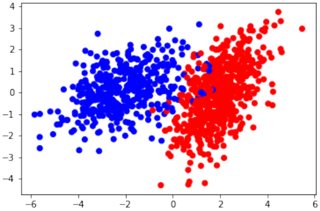


Parameter Estimation

- The collection of the parameters

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

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



- Learning algorithm: Maximum Likelihood Estimation (MLE)

Maximum Likelihood Estimation (MLE)

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

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

Maximum Likelihood Estimation (MLE)

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

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

Based on the chain rule of probability

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

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

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

Therefore

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

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

MLE: Bernoulli Distribution

Recall the definition of Bernoulli distribution, we have

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

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

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

Then, the value of α can be estimated from

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

therefore,

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

MLE: Gaussian Distribution

The definition of multi-variate Gaussian distribution

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

- 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

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

- 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

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

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

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

MLE: Gaussian Distribution

The definition of multi-variate Gaussian distribution

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

- 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 $\boldsymbol{\mu}_{+}$

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

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

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

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

Example: Parameter Estimation

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

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

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

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

Challenge of Parameter Estimation

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

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

- 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

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

Now, the multivariate Gaussian distribution can be rewritten with

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

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

Diagonal Covariance Matrix (II)

In other words

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

Diagonal Covariance Matrix (II)

In other words

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

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

Diagonal Covariance Matrix (II)

In other words

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

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

Diagonal Covariance Matrix (II)

In other words

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

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

Diagonal Covariance Matrix (II)

In other words

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

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

Latent Variable Models

EM Algorithm

Reference

Jurafsky, D. and Martin, J. (2019).
Speech and language processing.