Chapter 3

Maximum-Likelihood and Bayesian Parameter Estimation

Exercise

$p(\mathbf{x}|\omega_i) \sim N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^t\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_i| + \ln P(\omega_i)$$

$$g_i(\mathbf{x}) = \mathbf{x}^t\mathbf{W}_i\mathbf{x} + \mathbf{w}_i^t\mathbf{x} + w_{i0}$$

$$\mathbf{W}_i = -\frac{1}{2}\boldsymbol{\Sigma}_i^{-1} \quad \mathbf{w}_i = \boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i \quad w_{i0} = -\frac{1}{2}\boldsymbol{\mu}_i^t\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i - \frac{1}{2}\ln|\boldsymbol{\Sigma}_i| + \ln P(\omega_i)$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 3 \\ 6 \end{bmatrix}; \quad \boldsymbol{\Sigma}_1 = \begin{pmatrix} 1/2 & 0 \\ 0 & 2 \end{pmatrix} \text{ and } \boldsymbol{\mu}_2 = \begin{bmatrix} 3 \\ -2 \end{bmatrix}; \quad \boldsymbol{\Sigma}_2 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

Assumes equal prior probabilities,

What is the decision boundary?

Bayes Theorem for Classification

$$P(\omega_j|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_j)\cdot P(\omega_j)}{p(\mathbf{x})} \quad (1 \leq j \leq c)$$ (Bayes Formula)

To compute posterior probability $P(\omega_j|\mathbf{x})$, we need to know:

Prior probability: $P(\omega_j)$

Likelihood: $p(\mathbf{x}|\omega_j)$

The collection of training examples is composed of *c* data sets

$\mathcal{D}_j \ (1 \leq j \leq c)$

- □ Each example in $\mathcal{D}_j$ is drawn according to the class-conditional pdf, i.e. $p(\mathbf{x}|\omega_j)$
- □ Examples in $\mathcal{D}_j$ are *i.i.d.* random variables, i.e. **independent and identically distributed (独立同分布)**

Bayes Theorem for Classification (Cont.)

For prior probability: → **no difficulty**

$$P(\omega_j) = \frac{|\mathcal{D}_j|}{\sum_{i=1}^{c} |\mathcal{D}_i|}$$

(Here, $|\cdot|$ returns the **cardinality(势)**, i.e. number of elements, of a set)

For class-conditional pdf:

Ch. 3 ← □ **Case I:** $p(\mathbf{x}|\omega_j)$ has certain **parametric form**

$p(\mathbf{x}|\omega_j)$

e.g.: $p(\mathbf{x}|\omega_j) \sim N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ (**parameters:** $\boldsymbol{\theta}_j = \{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}$)

$\mathbf{x} \in \mathbf{R}^d \rightarrow \boldsymbol{\theta}_j$ contains "$d + d(d+1)/2$" $free$ parameters

To show the dependence of $p(\mathbf{x}|\omega_j)$ on $\boldsymbol{\theta}_j$ **explicitly:**

$$p(\mathbf{x}|\omega_j) \rightarrow p(\mathbf{x}|\omega_j, \boldsymbol{\theta}_j)$$

Ch. 4 ← □ **Case II:** $p(\mathbf{x}|\omega_j)$ doesn't have **parametric form**

Estimation Under Parametric Form

Parametric class-conditional pdf: $p(\mathbf{x}|\omega_j, \boldsymbol{\theta}_j)\ (1 \leq j \leq c)$

- □ **Assumption I: Maximum-Likelihood (ML) estimation (极大似然估计)**

View parameters as quantities whose values are **fixed but unknown** → Estimate parameter values by **maximizing the likelihood** (probability) of observing the actual training examples

- □ **Assumption II: Bayesian estimation (贝叶斯估计)**

View parameters as **random variables** having some known prior distribution → Observation of the actual training examples transforms parameters' **prior distribution into posterior distribution** (via Bayes theorem)

Bayesian vs Frequentist (Revisit)

Bob's Area
Alice's Area

- The Bayesian billiard game
  - Alice and Bob **can't** see the billiard table.
  - The judge rolls a ball down the table, and marks where it lands. Once this **mark** is in place, the judge rolls new balls.
  - If the ball lands to the left of the mark, Alice gets a point; if it lands to the right of the mark, Bob gets a point.
  - The first person to reach **6 points** wins the game.
  - Now say that Alice is leading with **5** points and Bob has **3** points.

What can be said about the chances of Bob to win the game?

Bayesian vs Frequentist (Revisit)

- The Frequentist Approach
  - 5 balls out of 8 balls fell on Alice's side
  - Maximum likelihood estimate of θ that balls land on Alice’s side:
    - $L(\theta) = p(y|\theta) = \binom{n}{y}\theta^{y}(1-\theta)^{n-y}$
    - $\log L(\theta) = y \log \theta + (n-y)\log(1-\theta) + C$
    - $\hat{\theta} = \frac{y}{n} = 5/8$
  - Assuming this maximum likelihood probability, we can compute the probability that Bob will win, which is given by:
    - $\mathrm{P}(Bob\ Wins) = (1-0.675)^{3} = 0.052734375$

Frequentist concludes that Bob got 5.2% chance of winning!

Maximum-Likelihood Estimation

Settings

Likelihood function for each category is governed by some **fixed but unknown** parameters, i.e. $p(\mathbf{x}|\omega_j, \boldsymbol{\theta}_j) \ (1 \leq j \leq c)$

Task: Estimate $\{\boldsymbol{\theta}_j\}_{j=1}^{c}$ from $\{\mathcal{D}_j\}_{j=1}^{c}$

A simplified treatment

Examples in $\mathcal{D}_j$ gives no information about $\boldsymbol{\theta}_i$ if $i \neq j$

⇓ ⇓ ⇓

Work with each category **separately** and therefore simplify the notations by dropping subscripts w.r.t. categories

without loss of generality: $\mathcal{D}_j \Longrightarrow \mathcal{D}$; $\boldsymbol{\theta}_j \Longrightarrow \boldsymbol{\theta}$

Maximum-Likelihood Estimation (Cont.)

$\mathbf{x}_k \sim p(\mathbf{x}|\boldsymbol{\theta})$

$(k = 1, \ldots, n)$

$\boldsymbol{\theta}$: Parameters to be estimated

$\mathcal{D}$: A set of *i.i.d.* examples $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$

The objective function

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{k=1}^{n} p(\boldsymbol{x}_k|\boldsymbol{\theta})$$

→ **The likelihood of $\boldsymbol{\theta}$ w.r.t. the set of observed examples**

The maximum-likelihood estimation

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\mathcal{D}|\boldsymbol{\theta})$$

→ **Intuitively, $\hat{\boldsymbol{\theta}}$ best agrees with the actually observed examples**

Maximum-Likelihood Estimation (Cont.)

Gradient Operator (梯度算子)

- ✓ Let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)^t \in \mathbf{R}^p$ be a *p*-dimensional vector
- ✓ Let $f : \mathbf{R}^p \rightarrow \mathbf{R}$ be *p*-variate real-valued function over $\boldsymbol{\theta}$

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} \equiv \begin{bmatrix} \frac{\partial}{\partial \theta_1} \\ \vdots \\ \frac{\partial}{\partial \theta_p} \end{bmatrix}$$

$$f(\boldsymbol{\theta}) = \theta_1^2 + 3\theta_1\theta_2$$

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} f = \begin{bmatrix} \frac{\partial f}{\partial \theta_1} \\ \frac{\partial f}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} 2\theta_1 + 3\theta_2 \\ 3\theta_1 \end{bmatrix}$$

$l(\boldsymbol{\theta}) = \ln p(\mathcal{D}|\boldsymbol{\theta})$ is named as the **log-likelihood function**

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\mathcal{D}|\boldsymbol{\theta}) \Longleftrightarrow \hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} l(\boldsymbol{\theta})$$

Maximum-Likelihood Estimation (Cont.)

$$l(\boldsymbol{\theta}) = \ln p(\mathcal{D}|\boldsymbol{\theta}) = \sum_{k=1}^{n} \ln p(\mathbf{x}_k|\boldsymbol{\theta})$$

$$\nabla_{\boldsymbol{\theta}} l = \nabla_{\boldsymbol{\theta}} \left(\sum_{k=1}^{n} \ln p(\mathbf{x}_k|\boldsymbol{\theta}) \right) = \sum_{k=1}^{n} \nabla_{\boldsymbol{\theta}} \ln p(\mathbf{x}_k|\boldsymbol{\theta})$$

p-dimensional vector with each component being a function over $\boldsymbol{\theta}$

p-variate real-valued function over $\boldsymbol{\theta}$ (not over $\mathbf{x}_k$)

Necessary conditions for ML estimate $\hat{\boldsymbol{\theta}}$

$$\nabla_{\boldsymbol{\theta}} l \,|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = \mathbf{0}$$ **(a set of *p* equations)**

The Gaussian Case: Unknown $\boldsymbol{\mu}$

$\mathbf{x}_k \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

$(k = 1, \ldots, n)$

suppose $\boldsymbol{\Sigma}$ is known → $\boldsymbol{\theta} = \{\boldsymbol{\mu}\}$

$$p(\mathbf{x}_k|\boldsymbol{\mu}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1} (\mathbf{x}_k - \boldsymbol{\mu})\right]$$

$$\ln p(\mathbf{x}_k|\boldsymbol{\mu}) = -\frac{1}{2}\ln\left[(2\pi)^d|\boldsymbol{\Sigma}|\right] - \frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu})^t\boldsymbol{\Sigma}^{-1}(\mathbf{x}_k - \boldsymbol{\mu})$$

$$= -\frac{1}{2}\ln\left[(2\pi)^d|\boldsymbol{\Sigma}|\right] - \frac{1}{2}\mathbf{x}_k^t\boldsymbol{\Sigma}^{-1}\mathbf{x}_k + \boldsymbol{\mu}^t\boldsymbol{\Sigma}^{-1}\mathbf{x}_k - \frac{1}{2}\boldsymbol{\mu}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$$

$$\boldsymbol{\nabla}_{\boldsymbol{\mu}} \ln p(\mathbf{x}_k|\boldsymbol{\mu}) = \boldsymbol{\Sigma}^{-1}(\mathbf{x}_k - \boldsymbol{\mu})$$

The Gaussian Case: Unknown $\boldsymbol{\mu}$ (Cont.)

$$l(\boldsymbol{\mu}) = \sum_{k=1}^{n} \ln p(\mathbf{x}_k|\boldsymbol{\mu})$$

$$\nabla_{\boldsymbol{\mu}} \ln p(\mathbf{x}_k|\boldsymbol{\mu}) = \boldsymbol{\Sigma}^{-1}(\mathbf{x}_k - \boldsymbol{\mu})$$

$$\nabla_{\boldsymbol{\mu}} l = \sum_{k=1}^{n} \boldsymbol{\Sigma}^{-1}(\mathbf{x}_k - \boldsymbol{\mu})$$

$\nabla_{\boldsymbol{\mu}} l = \mathbf{0}$ (necessary condition for ML estimate $\hat{\boldsymbol{\mu}}$)

$$\sum_{k=1}^{n} \boldsymbol{\Sigma}^{-1}(\mathbf{x}_k - \hat{\boldsymbol{\mu}}) = \mathbf{0}$$

Multiply $\boldsymbol{\Sigma}$ on both sides

$$\sum_{k=1}^{n} (\mathbf{x}_k - \hat{\boldsymbol{\mu}}) = \mathbf{0}$$

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k$$

Intuitive result

ML estimate for the unknown $\boldsymbol{\mu}$ is just the arithmetic average of training samples – ***sample mean***

The Gaussian Case: Unknown $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$

$$\mathbf{x}_k \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$(k = 1, \ldots, n)$$

$$\boldsymbol{\mu} \text{ and } \boldsymbol{\Sigma} \text{ unknown} \Rightarrow \boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$$

Consider *univariate* case

$$p(x_k|\boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \qquad \left(\boldsymbol{\theta} = \begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix} = \begin{bmatrix}\mu\\ \sigma^2\end{bmatrix}\right)$$

$$\Rightarrow \ln p(x_k|\boldsymbol{\theta}) = -\frac{1}{2}\ln 2\pi\theta_2 - \frac{1}{2\theta_2}(x_k - \theta_1)^2$$

$$\Rightarrow \nabla_{\boldsymbol{\theta}} \ln p(x_k|\boldsymbol{\theta}) = \begin{bmatrix} \frac{1}{\theta_2}(x_k - \theta_1) \\ -\frac{1}{2\theta_2} + \frac{(x_k-\theta_1)^2}{2\theta_2^2} \end{bmatrix}$$

The Gaussian Case: Unknown $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ (Cont.)

$$l(\boldsymbol{\theta}) = \sum_{k=1}^{n} \ln p(x_k|\boldsymbol{\theta})$$

$$\nabla_{\boldsymbol{\theta}} \ln p(x_k|\boldsymbol{\theta}) = \begin{bmatrix} \frac{1}{\theta_2}(x_k - \theta_1) \\ -\frac{1}{2\theta_2} + \frac{(x_k-\theta_1)^2}{2\theta_2^2} \end{bmatrix}$$

$$\nabla_{\boldsymbol{\theta}} l = \begin{bmatrix} \sum_{k=1}^{n} \frac{1}{\theta_2}(x_k - \theta_1) \\ \sum_{k=1}^{n} \left(-\frac{1}{2\theta_2} + \frac{(x_k-\theta_1)^2}{2\theta_2^2}\right) \end{bmatrix}$$

$\nabla_{\boldsymbol{\theta}} l = 0$ (necessary condition for ML estimate $\hat{\theta}_1$ and $\hat{\theta}_2$)

$$\sum_{k=1}^{n} \frac{1}{\hat{\theta}_2}(x_k - \hat{\theta}_1) = 0$$

$$-\sum_{k=1}^{n} \frac{1}{\hat{\theta}_2} + \sum_{k=1}^{n} \frac{(x_k - \hat{\theta}_1)^2}{\hat{\theta}_2^2} = 0$$

The Gaussian Case: Unknown $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ (Cont.)

$$\sum_{k=1}^{n} \frac{1}{\hat{\theta}_2}(x_k - \hat{\theta}_1) = 0 \Longrightarrow \sum_{k=1}^{n}(x_k - \hat{\theta}_1) = 0 \Longrightarrow \hat{\theta}_1 = \frac{1}{n}\sum_{k=1}^{n} x_k$$

$$-\sum_{k=1}^{n} \frac{1}{\hat{\theta}_2} + \sum_{k=1}^{n} \frac{(x_k - \hat{\theta}_1)^2}{\hat{\theta}_2^2} = 0 \Longrightarrow \hat{\theta}_2 = \frac{1}{n}\sum_{k=1}^{n}(x_k - \hat{\theta}_1)^2$$

ML estimate in *univariate* case

$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k \qquad \hat{\sigma}^2 = \frac{1}{n}\sum_{k=1}^{n}(x_k - \hat{\mu})^2$$

The Gaussian Case: Unknown $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ (Cont.)

Intuitive result as well!

ML estimate in *multivariate* case

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_k$$

→ *Arithmetic average* of *n* vectors $\mathbf{x}_k$

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{k=1}^{n}(\mathbf{x}_k - \hat{\boldsymbol{\mu}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}})^t$$

→ *Arithmetic average* of *n* matrices $(\mathbf{x}_k - \hat{\boldsymbol{\mu}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}})^t$

Bayesian vs Frequentist (Revisit)

- The Bayesian Approach
  - Prior distributions: $\theta \sim Uniform(0,1)$
  - $\mathbb{E}(Bob\,wins) = \int_0^1 (1-\theta)^3 P(\theta|A=5, B=3)d\theta$
  - $P(\theta|A=5, B=3) = \frac{P(\theta)P(A=5,B=3|\theta)}{\int_0^1 P(\theta)P(A=5,B=3|\theta)d\theta}$
  - $P(A=5, B=3|\theta) = \binom{8}{5}\theta^5(1-\theta)^3$, $P(\theta) = 1$
  - $\mathbb{E}(Bob\,wins) = \frac{\int_0^1 (1-\theta)^6\theta^5 d\theta}{\int_0^1 (1-\theta)^3\theta^5 d\theta} = \frac{5!6!/12!}{5!3!/9!} = \frac{1}{11}$
  - Without knowing the Bayesian probability:
    - $\mathbb{E}(Bob\,Wins) = 0.091$

$$\int_0^1 p^{m-1}(1-p)^{n-1}\,dp = \frac{\Gamma(n)\Gamma(m)}{\Gamma(m+n)}$$

$$\Gamma(n+1) = n!$$

Bayesian concludes that Bob got 9.1% chance of winning!

Bayesian Estimation

Settings

- □ The **parametric form** of the likelihood function for each category is known $p(\mathbf{x}|\omega_j, \boldsymbol{\theta}_j) \ (1 \leq j \leq c)$
- □ However, $\boldsymbol{\theta}_j$ is considered to be **random variables** instead of being fixed (but unknown) values

In this case, we can no longer make a single ML estimate $\hat{\boldsymbol{\theta}}_j$ and then infer $P(\omega_j|\mathbf{x})$ based on $P(\omega_j)$ and $p(\mathbf{x}|\omega_j, \hat{\boldsymbol{\theta}}_j)$

How can we proceed under this situation

Fully exploit training examples!

$$P(\omega_j|\mathbf{x}) \Rightarrow P(\omega_j|\mathbf{x}, \mathcal{D}^*)$$

$$(\mathcal{D}^* = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \cdots \cup \mathcal{D}_c)$$

Bayesian Estimation (Cont.)

$$P(\omega_j|\mathbf{x}, \mathcal{D}^*) = \frac{p(\omega_j, \mathbf{x}, \mathcal{D}^*)}{p(\mathbf{x}, \mathcal{D}^*)} = \frac{p(\omega_j, \mathbf{x}, \mathcal{D}^*)}{\sum_{i=1}^{c} p(\omega_i, \mathbf{x}, \mathcal{D}^*)}$$

$$p(\omega_j, \mathbf{x}, \mathcal{D}^*) = p(\mathcal{D}^*) \cdot p(\omega_j, \mathbf{x}|\mathcal{D}^*) = p(\mathcal{D}^*) \cdot P(\omega_j|\mathcal{D}^*) \cdot p(\mathbf{x}|\omega_j, \mathcal{D}^*)$$

$$P(\omega_j|\mathbf{x}, \mathcal{D}^*) = \frac{p(\mathcal{D}^*) \cdot P(\omega_j|\mathcal{D}^*) \cdot p(\mathbf{x}|\omega_j, \mathcal{D}^*)}{p(\mathcal{D}^*) \cdot \sum_{i=1}^{c} P(\omega_i|\mathcal{D}^*) \cdot p(\mathbf{x}|\omega_i, \mathcal{D}^*)}$$

$$= \frac{P(\omega_j|\mathcal{D}^*) \cdot p(\mathbf{x}|\omega_j, \mathcal{D}^*)}{\sum_{i=1}^{c} P(\omega_i|\mathcal{D}^*) \cdot p(\mathbf{x}|\omega_i, \mathcal{D}^*)}$$

Eq.22 [pp.91]

Two assumptions

$P(\omega_j|\mathcal{D}^*) = P(\omega_j)$

$p(\mathbf{x}|\omega_j, \mathcal{D}^*) = p(\mathbf{x}|\omega_j, \mathcal{D}_j)$

$$= \frac{P(\omega_j) \cdot p(\mathbf{x}|\omega_j, \mathcal{D}_j)}{\sum_{i=1}^{c} P(\omega_i) \cdot p(\mathbf{x}|\omega_i, \mathcal{D}_i)}$$

Eq.23 [pp.91]

Bayesian Estimation (Cont.)

$$P(\omega_j|\mathbf{x},\mathcal{D}^*) = \frac{P(\omega_j)\cdot p(\mathbf{x}|\omega_j,\mathcal{D}_j)}{\sum_{i=1}^{c} P(\omega_i)\cdot p(\mathbf{x}|\omega_i,\mathcal{D}_i)}$$

Key problem

Determine $p(\mathbf{x}|\omega_j,\mathcal{D}_j)$

Treat each class independently → Simplify the *class-conditional pdf* notation $p(\mathbf{x}|\omega_j,\mathcal{D}_j)$ as $p(\mathbf{x}|\mathcal{D})$

$$p(\mathbf{x}|\mathcal{D}) = \int p(\mathbf{x},\boldsymbol{\theta}|\mathcal{D})\,d\boldsymbol{\theta}$$

($\boldsymbol{\theta}$: random variables w.r.t. parametric form)

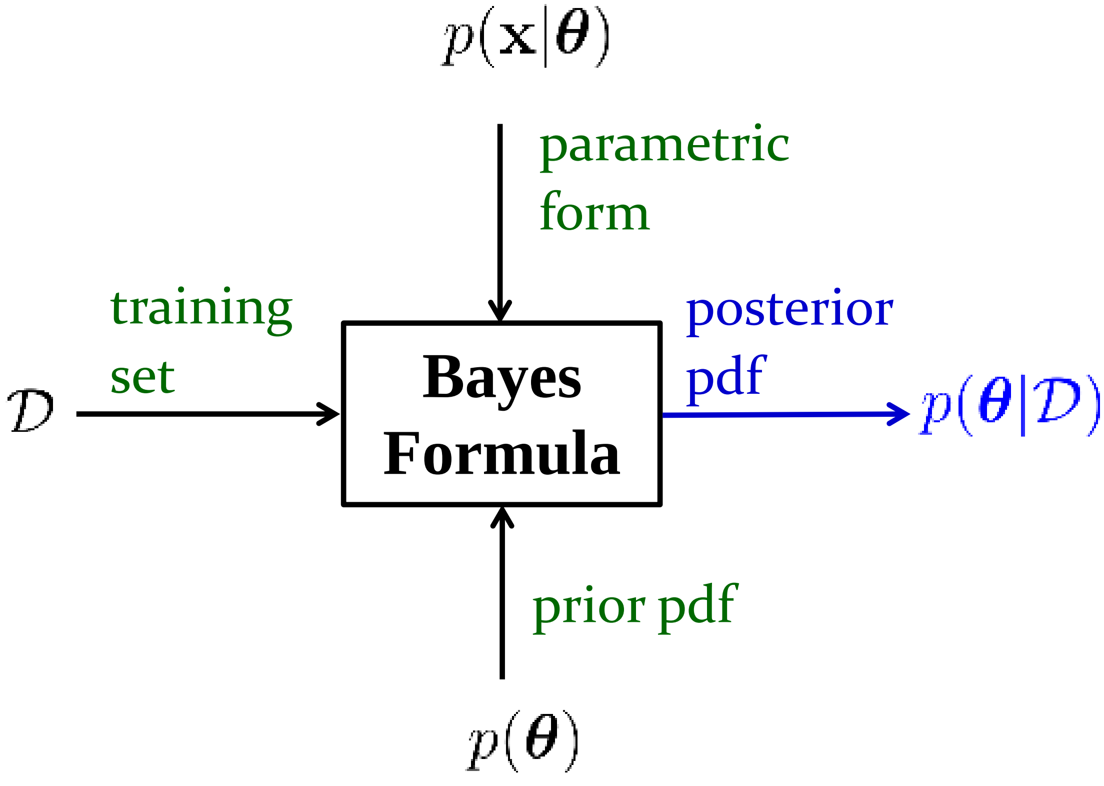
$$= \int p(\mathbf{x}|\boldsymbol{\theta},\mathcal{D})\,p(\boldsymbol{\theta}|\mathcal{D})\,d\boldsymbol{\theta}$$

$$= \int p(\mathbf{x}|\boldsymbol{\theta})\,p(\boldsymbol{\theta}|\mathcal{D})\,d\boldsymbol{\theta}$$

($\mathbf{x}$ is independent of $\mathcal{D}$ given $\boldsymbol{\theta}$)

Bayesian Estimation: The General Procedure

Phase I: *prior pdf* ➔ *posterior pdf* (for $\boldsymbol{\theta}$)



$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\boldsymbol{\theta}, \mathcal{D})}{p(\mathcal{D})}$$

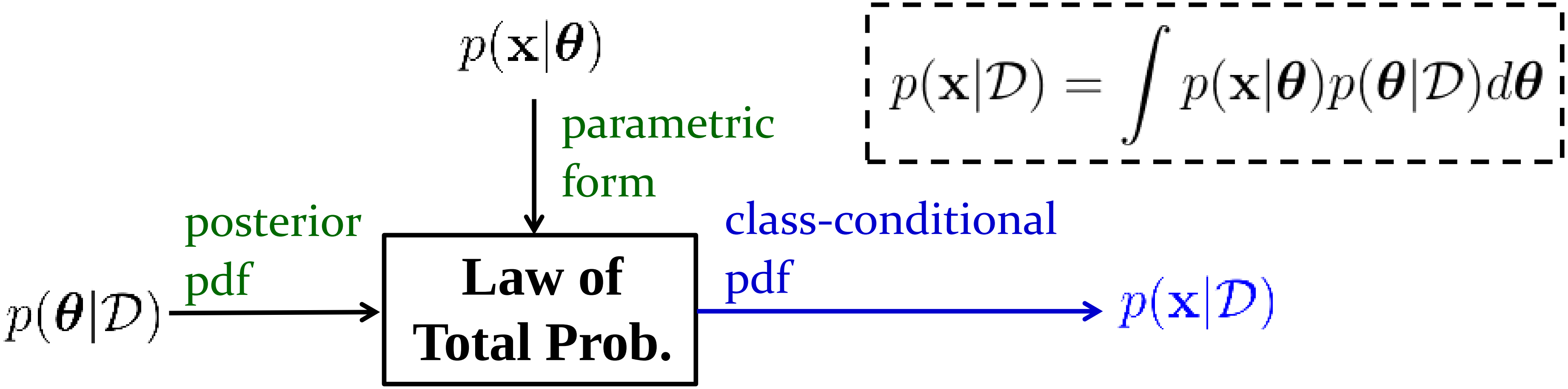
$$= \frac{p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})}{\int p(\boldsymbol{\theta}, \mathcal{D})d\boldsymbol{\theta}}$$

$$= \frac{p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})}{\int p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})d\boldsymbol{\theta}}$$

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{k=1}^{n} p(\mathbf{x}_k|\boldsymbol{\theta})$$

Bayesian Estimation: The General Procedure

Phase II: *posterior pdf* (for $\boldsymbol{\theta}$) ➔ *class-conditional pdf* (for $\mathbf{x}$)



$$p(\mathbf{x}|\mathcal{D}) = \int p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}$$

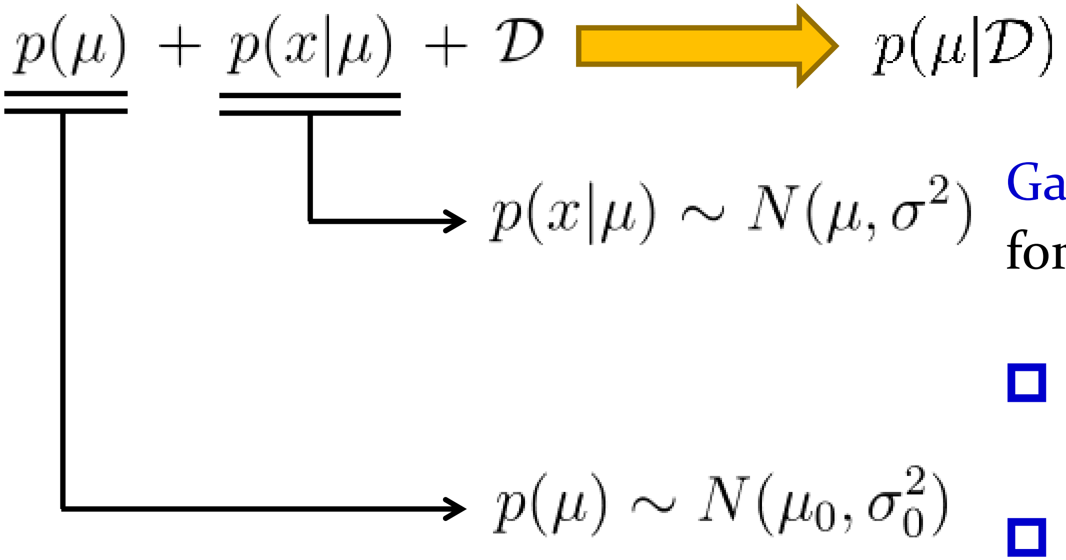
$p(\mathbf{x}|\boldsymbol{\theta})$
parametric form
posterior pdf
$p(\boldsymbol{\theta}|\mathcal{D})$
Law of Total Prob.
class-conditional pdf
$p(\mathbf{x}|\mathcal{D})$

Phase III: $$P(\omega_j|\mathbf{x}, \mathcal{D}^*) = \frac{P(\omega_j) \cdot p(\mathbf{x}|\omega_j, \mathcal{D}_j)}{\sum_{i=1}^{c} P(\omega_i) \cdot p(\mathbf{x}|\omega_i, \mathcal{D}_i)}$$

The Gaussian Case: Unknown $\boldsymbol{\mu}$

Consider *univariate* case: $\boldsymbol{\theta} = \{\mu\}$ (σ^2 is known)

Phase I: *prior pdf* ➔ *posterior pdf* (for $\boldsymbol{\theta}$)



$$\underline{\underline{p(\mu)}} + \underline{\underline{p(x|\mu)}} + \mathcal{D} \Longrightarrow p(\mu|\mathcal{D})$$

$$p(x|\mu) \sim N(\mu, \sigma^2)$$

Gaussian parametric form

$$p(\mu) \sim N(\mu_0, \sigma_0^2)$$

- □ Prior pdf still takes Gaussian form
- □ Other form of prior pdf could be assumed as well

How would $p(\mu|\mathcal{D})$ look like in this case?

The Gaussian Case: Unknown $\boldsymbol{\mu}$ (Cont.)

$$p(\mu|\mathcal{D}) = \frac{p(\mu,\mathcal{D})}{p(\mathcal{D})} = \frac{p(\mu)p(\mathcal{D}|\mu)}{\int p(\mu)p(\mathcal{D}|\mu)\,d\mu}$$

$$= \alpha\, p(\mu)\, p(\mathcal{D}|\mu)$$

($\int p(\mu)p(\mathcal{D}|\mu)\,d\mu$ is a constant not related to μ)

$$= \alpha\, p(\mu) \prod_{k=1}^{n} p(x_k|\mu)$$

(examples in $\mathcal{D}$ are *i.i.d.*)

$$p(\mu) \sim N(\mu_0, \sigma_0^2) \qquad p(x|\mu) \sim N(\mu, \sigma^2)$$

$$p(\mu) = \frac{1}{\sqrt{2\pi}\sigma_0} \exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_0}{\sigma_0}\right)^2\right] \qquad p(x_k|\mu) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x_k-\mu}{\sigma}\right)^2\right]$$

The Gaussian Case: Unknown $\boldsymbol{\mu}$ (Cont.)

$p(\mu|\mathcal{D})$ is an exponential function of a quadratic function of μ ⇨ $p(\mu|\mathcal{D})$ is a normal pdf as well

$$p(\mu|\mathcal{D}) = \alpha\, p(\mu) \prod_{k=1}^{n} p(x_k|\mu)$$

$$= \alpha \cdot \frac{1}{\sqrt{2\pi}\sigma_0} \exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_0}{\sigma_0}\right)^2\right] \cdot \prod_{k=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x_k-\mu}{\sigma}\right)^2\right]$$

$$= \alpha' \cdot \exp\left[-\frac{1}{2}\left(\left(\frac{\mu-\mu_0}{\sigma_0}\right)^2 + \sum_{k=1}^{n}\left(\frac{\mu-x_k}{\sigma}\right)^2\right)\right]$$

$p(\mu|\mathcal{D}) \sim N(\mu_n, \sigma_n^2)$

$$= \alpha'' \cdot \exp\left[-\frac{1}{2}\left[\left(\frac{n}{\sigma^2}+\frac{1}{\sigma_0^2}\right)\mu^2 - 2\left(\frac{1}{\sigma^2}\sum_{k=1}^{n} x_k + \frac{\mu_0}{\sigma_0^2}\right)\mu\right]\right]$$

The Gaussian Case: Unknown $\boldsymbol{\mu}$ (Cont.)

$$p(\mu|\mathcal{D}) = \alpha'' \cdot \exp\left[-\frac{1}{2}\left[\boxed{\left(\frac{n}{\sigma^2}+\frac{1}{\sigma_0^2}\right)}\mu^2 - 2\boxed{\left(\frac{1}{\sigma^2}\sum_{k=1}^{n}x_k+\frac{\mu_0}{\sigma_0^2}\right)}\mu\right]\right]$$

$$p(\mu|\mathcal{D}) = \frac{1}{\sqrt{2\pi}\sigma_n}\exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_n}{\sigma_n}\right)^2\right] = \alpha'' \cdot \exp\left[-\frac{1}{2}\left[\boxed{\frac{1}{\sigma_n^2}}\mu^2 - 2\boxed{\frac{\mu_n}{\sigma_n^2}}\mu\right]\right]$$

Equating the coefficients in both form:

$$\frac{1}{\sigma_n^2} = \frac{n}{\sigma^2}+\frac{1}{\sigma_0^2}$$

$$\frac{\mu_n}{\sigma_n^2} = \frac{1}{\sigma^2}\sum_{k=1}^{n}x_k+\frac{\mu_0}{\sigma_0^2}$$

$$\sigma_n^2 = \frac{\sigma^2\sigma_0^2}{n\sigma_0^2+\sigma^2}$$

$$\mu_n = \frac{\sigma_n^2}{\sigma^2}\sum_{k=1}^{n}x_k+\frac{\sigma_n^2}{\sigma_0^2}\mu_0$$

The Gaussian Case: Unknown $\boldsymbol{\mu}$ (Cont.)

Phase II: *posterior pdf* (for $\boldsymbol{\theta}$) ➔ *class-conditional pdf* (for **x**)

$$\underline{\underline{p(\mu|\mathcal{D})}} + \underline{\underline{p(x|\mu)}} \Longrightarrow p(x|\mathcal{D})$$

$$p(x|\mu) \sim N(\mu, \sigma^2)$$

$$p(\mu|\mathcal{D}) \sim N(\mu_n, \sigma_n^2)$$

How would $p(x|\mathcal{D})$ look like in this case?

$$\sigma_n^2 = \frac{\sigma^2\sigma_0^2}{n\sigma_0^2 + \sigma^2}$$

$$\mu_n = \frac{\sigma_n^2}{\sigma^2}\sum_{k=1}^{n} x_k + \frac{\sigma_n^2}{\sigma_0^2}\mu_0$$

The Gaussian Case: Unknown $\boldsymbol{\mu}$ (Cont.)

Then, phase III follows naturally for prediction

$$p(x|\mathcal{D}) = \int p(x|\mu)p(\mu|\mathcal{D})d\mu$$

Eq.25 [pp.92]

$$= \int \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]\frac{1}{\sqrt{2\pi}\sigma_n}\exp\left[-\frac{1}{2}\left(\frac{\mu-\mu_n}{\sigma_n}\right)^2\right]d\mu$$

$$= \beta\cdot\exp\left[-\frac{1}{2}\frac{(x-\mu_n)^2}{\sigma^2+\sigma_n^2}\right]$$

Eq.36 [pp.95]

$p(x|\mathcal{D})$ is an exponential function of a quadratic function of x → $p(x|\mathcal{D})$ is a normal pdf as well

$p(x|\mathcal{D}) \sim N(\mu_n, \sigma^2+\sigma_n^2)$

The Gaussian Case: Unknown $\boldsymbol{\mu}$ (Multivariate)

$\boldsymbol{\theta} = \{\boldsymbol{\mu}\}$ ($\boldsymbol{\Sigma}$ is known) ⇨

$$p(\mathbf{x}|\boldsymbol{\mu}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$p(\boldsymbol{\mu}) \sim N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$$

$$p(\boldsymbol{\mu}|\mathcal{D}) \sim N(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n) \qquad p(\mathbf{x}|\mathcal{D}) \sim N(\boldsymbol{\mu}_n, \boldsymbol{\Sigma} + \boldsymbol{\Sigma}_n)$$

$$\boldsymbol{\mu}_n = \boldsymbol{\Sigma}_0 \left(\boldsymbol{\Sigma}_0 + \frac{1}{n}\boldsymbol{\Sigma} \right)^{-1} \frac{1}{n} \sum_{k=1}^{n} \mathbf{x}_k + \frac{1}{n}\boldsymbol{\Sigma} \left(\boldsymbol{\Sigma}_0 + \frac{1}{n}\boldsymbol{\Sigma} \right)^{-1} \boldsymbol{\mu}_0$$

$$\boldsymbol{\Sigma}_n = \boldsymbol{\Sigma}_0 \left(\boldsymbol{\Sigma}_0 + \frac{1}{n}\boldsymbol{\Sigma} \right)^{-1} \frac{1}{n}\boldsymbol{\Sigma}$$

A Few Notes on Parametric Techniques

ML estimation vs. Bayes estimation

- *Infinite examples* ML estimation = Bayes estimation
- *Complexity* ML estimation < Bayes estimation
- *Interpretability* ML estimation > Bayes estimation
- *Prior knowledge* ML estimation < Bayes estimation

Source of classification error

Bayes error + Model error + Estimation error

Related Topic I

Hidden Markov Model

Markov Model

- a **Markov model** is a **stochastic** model used to model **pseudo-randomly** changing systems.
- a **Markov chain** is a stochastic model describing a sequence of events in which the probability of each event depends only on the state of the previous event.

Andrey Andreyevich Markov
1856-1922
Russian Mathematics

Markov first worked as a mathematician at the St. Petersburg University. Later during 1908, he quitted being a lecturer became a teacher at a high school.

Markov Model (Cont.)

Notations

$\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}$: A set of *c* possible states

$\boldsymbol{\omega}^T = \{\omega(1), \omega(2), \ldots, \omega(T)\}$: A state sequence of length *T*, where $\omega(t) \in \Omega$ $(1 \leq t \leq T)$

e.g.: $\boldsymbol{\omega}^6 = \{\omega_1, \omega_4, \omega_2, \omega_2, \omega_1, \omega_4\}$

$\mathbf{A} = [a_{ij}]_{c \times c}$: The *transition probability* matrix

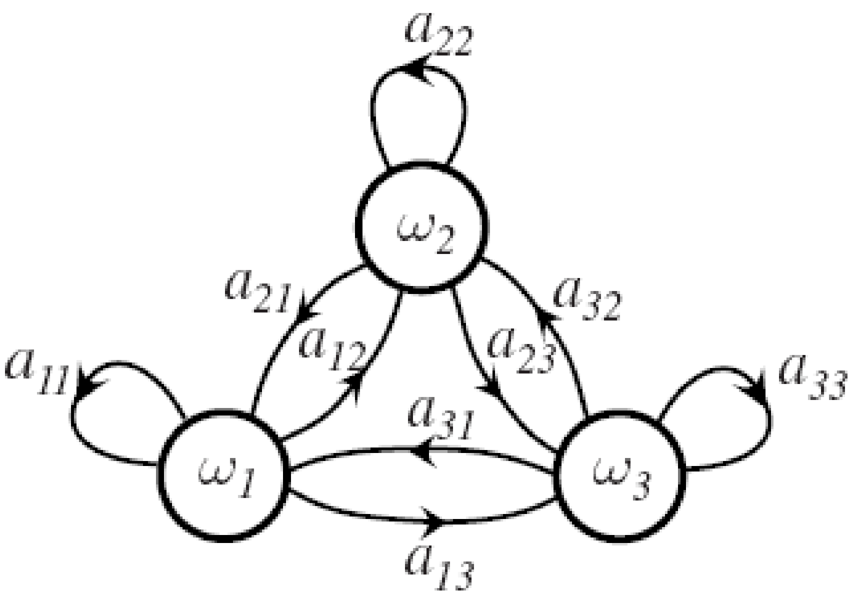
$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1c} \\ a_{21} & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ a_{c1} & \cdots & \cdots & a_{cc} \end{bmatrix}$$

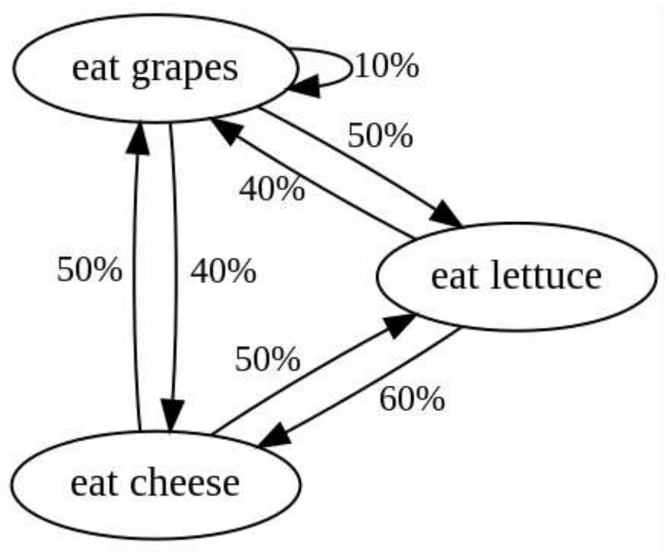
$$a_{ij} = P(\omega(t+1) = \omega_j \mid \omega(t) = \omega_i)$$
$$= P(\omega_j \mid \omega_i)$$

(time-independent) probability of transferring from state ω_i to state ω_j

$\sum_{j=1}^{c} a_{ij} = 1$, and in general $a_{ij} \neq a_{ji}$

Markov Model (Cont.)



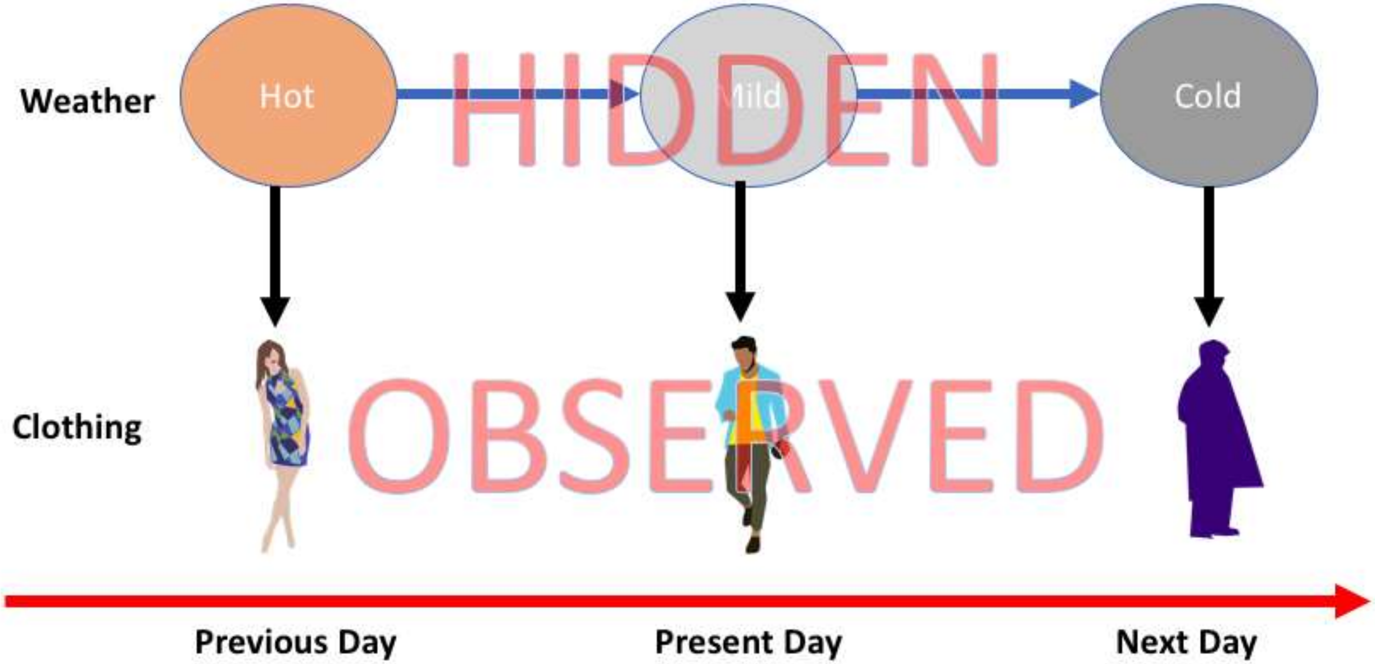


$$\boldsymbol{\omega}^T = \{\omega(1), \omega(2), \ldots, \omega(T)\}:$$

$$P(\boldsymbol{\omega}^T) = \prod_{t=1}^{T} P(\omega(t) \mid \omega(1), \ldots, \omega(t-1)) \quad \textit{(chain rule)}$$

$$= \prod_{t=1}^{T} P(\omega(t) \mid \omega(t-1)) \quad \textit{(first-order assumption)}$$

Hidden Markov Model (HMM)



Basic assumptions

- □ The state at each step is invisible
- □ The invisible state emits one visible symbol at each step

Hidden Markov Model (HMM)

Basic assumptions

- The state at each step is invisible
- The invisible state emits one visible symbol at each step

A few more notations

$\mathcal{V} = \{v_1, v_2, \ldots, v_K\}$: A set of *K* possible symbols

$\mathbf{V}^T = \{v(1), v(2), \ldots, v(T)\}$: An observed symbol sequence of length *T*, where $v(t) \in \mathcal{V} \ (1 \leq t \leq T)$

$\mathbf{B} = [b_{jk}]_{c \times K}$: The *observation symbol probability* matrix

$$\begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1K} \\ \cdots & \cdots & \cdots & \cdots \\ b_{c1} & \cdots & \cdots & b_{cK} \end{bmatrix}$$

$b_{jk} = P(v_k \mid \omega_j), \ \sum_{k=1}^{K} b_{jk} = 1$

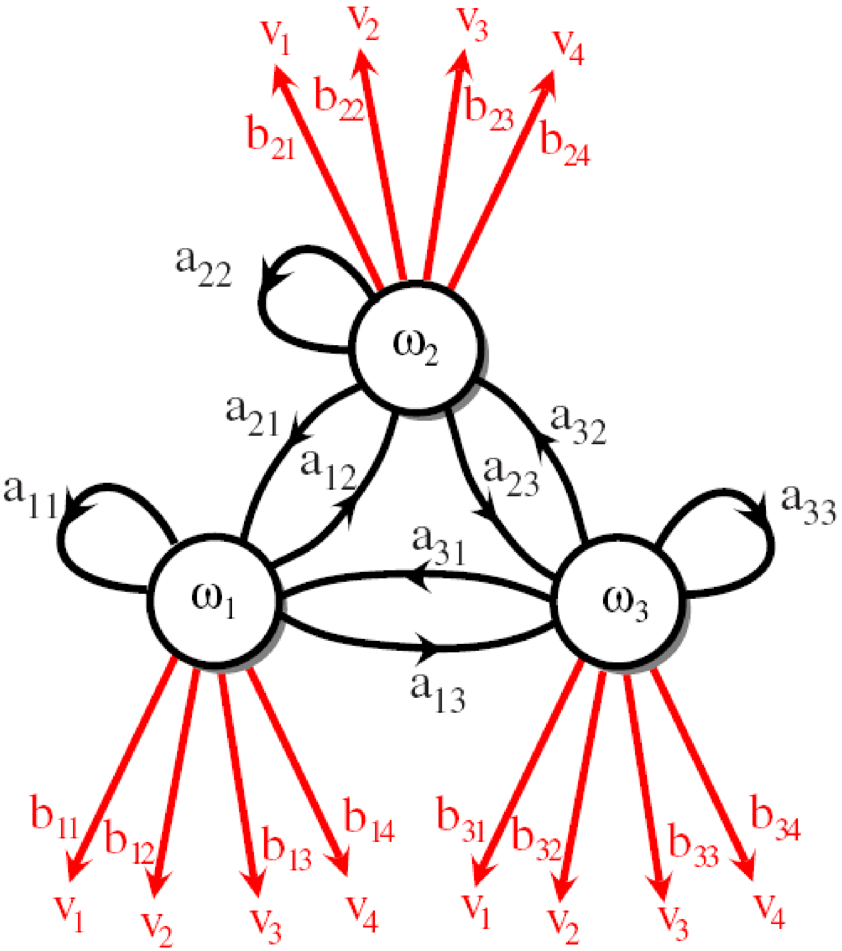
probability of emitting symbol v_k *at state* ω_j

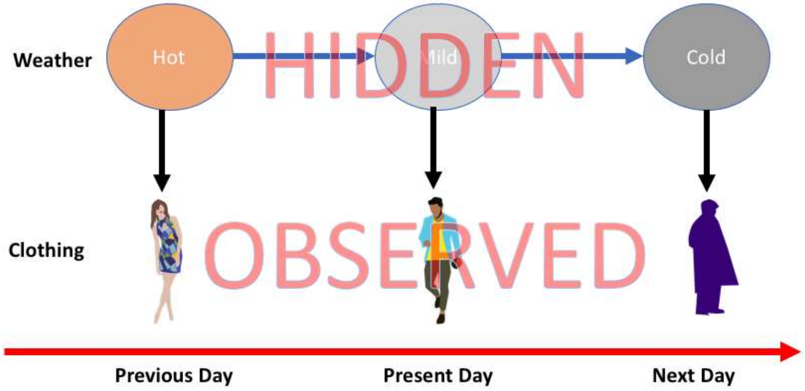
$\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_c)$: The initial state probability

$\pi_j = P(\omega(1) = \omega_j)$

Hidden Markov Model (Cont.)

State transition diagram





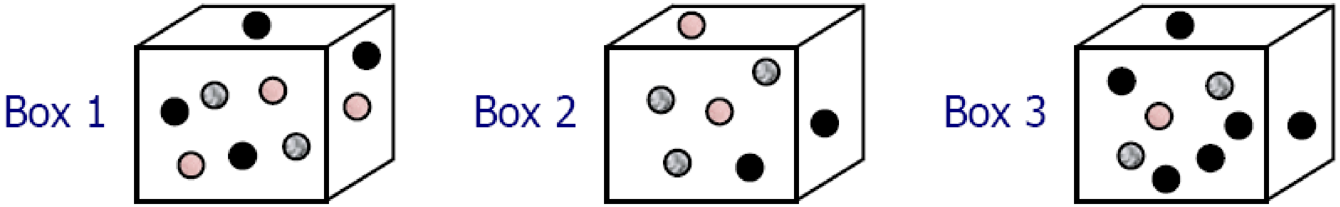
$$P(\mathbf{V}^T \mid \boldsymbol{\omega}^T) = \prod_{t=1}^{T} P(v(t) \mid \omega(t))$$

$$= \prod_{t=1}^{T} b_{\omega(t)v(t)}$$

The probability of emitting one symbol at each step only depends on the state at that step

Hidden Markov Model (Cont.)

An illustrative example



Hidden state: box

Visible symbol: ball

Observation symbol probability: $P($●$\mid box\ i)$, $P($◍$\mid box\ i)$, $P($○$\mid box\ i)$

Observed symbol sequence: ● ● ◍ ○ ◍ ● ○ ○ ● ◍

Given the observed symbol sequence, what are the central problems in HMM?

Hidden Markov Model (Cont.)

Three central problems in HMM

$\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$: the complete set of HMM parameters

$\mathbf{V}^T$: the observed symbol sequence

Evaluation

Given $\boldsymbol{\theta}$, determine the probability of generating $\mathbf{V}^T$

to evaluate $P(\mathbf{V}^T \mid \boldsymbol{\theta})$

Learning

Given $\mathbf{V}^T$, determine model parameters $\boldsymbol{\theta}$

to identify $\boldsymbol{\theta}$ *which maximizes* $P(\mathbf{V}^T \mid \boldsymbol{\theta})$

Decoding

Given $\boldsymbol{\theta}$ and $\mathbf{V}^T$, determine the most likely hidden state sequence

to identify $\boldsymbol{\omega}^T$ *which maximizes* $P(\boldsymbol{\omega}^T \mid \mathbf{V}^T, \boldsymbol{\theta})$

The Evaluation Problem for HMM

A straightforward evaluation

$$P(\mathbf{V}^T \mid \boldsymbol{\theta}) = \sum_{\boldsymbol{\omega}^T} P(\mathbf{V}^T \mid \boldsymbol{\omega}^T, \boldsymbol{\theta}) P(\boldsymbol{\omega}^T \mid \boldsymbol{\theta})$$

$$P(\boldsymbol{\omega}^T \mid \boldsymbol{\theta}) = \prod_{t=1}^{T} a_{\omega(t-1)\omega(t)}$$ (*with abuse of notation*: $a_{\omega(0)\omega(1)} = \pi_{\omega(1)}$)

$$P(\mathbf{V}^T \mid \boldsymbol{\omega}^T, \boldsymbol{\theta}) = \prod_{t=1}^{T} b_{\omega(t)v(t)}$$

$$P(\mathbf{V}^T \mid \boldsymbol{\theta}) = \sum_{\boldsymbol{\omega}^T} \prod_{t=1}^{T} a_{\omega(t-1)\omega(t)} b_{\omega(t)v(t)}$$

Computational complexity: $\mathcal{O}(c^T \cdot T)$!

e.g.: *c*=10, *T*=20 ➔ ~ 10^{21} calculations

Infeasible!

The Evaluation Problem for HMM (Cont.)

HMM forward algorithm

$$P(\mathbf{V}^T \mid \boldsymbol{\theta}) = \sum_{j=1}^{c} \alpha_j(T)$$

Let $\alpha_j(t) = P(v(1), v(2), \ldots, v(t), \omega(t) = \omega_j \mid \boldsymbol{\theta})$

the probability of being in hidden state ω_j *at step t and having generated the first t symbols of* $\mathbf{V}^T$

Then, $\alpha_j(t)\ (1 \leq j \leq c, 1 \leq t \leq T)$ can be calculated recursively as:

$$\alpha_j(1) = P(v(1), \omega(1) = \omega_j \mid \boldsymbol{\theta}) = P(\omega(1) = \omega_j \mid \boldsymbol{\theta}) \cdot P(v(1) \mid \omega_j, \boldsymbol{\theta})$$

$$= \pi_j b_{jv(1)}$$

$$\alpha_j(t) = \sum_{i=1}^{c} P(v(1), \ldots, v(t-1), \omega(t-1) = \omega_i, v(t), \omega(t) = \omega_j \mid \boldsymbol{\theta})$$

$$= \sum_{i=1}^{c} P(v(1), \ldots, v(t-1), \omega(t-1) = \omega_i \mid \boldsymbol{\theta}) \cdot P(\omega_j \mid \omega_i, \boldsymbol{\theta}) \cdot P(v(t) \mid \omega_j, \boldsymbol{\theta})$$

$$= \left[\sum_{i=1}^{c} \alpha_i(t-1) a_{ij}\right] b_{jv(t)}$$

The Evaluation Problem for HMM (Cont.)

$\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$: the complete set of HMM parameters

$\mathbf{V}^T$: the observed symbol sequence

to evaluate

$P(\mathbf{V}^T \mid \boldsymbol{\theta})$

Pseudo-code for HMM forward algorithm

```
Initialize $t = 1$ and $\alpha_j(t) = \pi_j b_{jv(t)}$ $(1 \le j \le c)$
For $t = 2$ to $T$
    For $j = 1$ to $c$
        $\alpha_j(t) = \left[\sum_{i=1}^{c} \alpha_i(t-1) a_{ij}\right] b_{jv(t)}$
    End
End
Return $P(\mathbf{V}^T \mid \boldsymbol{\theta}) = \sum_{j=1}^{c} \alpha_j(T)$
```

Computational complexity

$\mathcal{O}(c^T \cdot T)$

$\mathcal{O}(c^2 \cdot T)$

The Evaluation Problem for HMM (Cont.)

$\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$: the complete set of HMM parameters

$\mathbf{V}^T$: the observed symbol sequence

to evaluate $P(\mathbf{V}^T \mid \boldsymbol{\theta})$

Let $\alpha_j(t) = P(v(1), v(2), \ldots, v(t), \omega(t) = \omega_j \mid \boldsymbol{\theta})$

the probability of being in hidden state ω_j *at step t and having generated the first t symbols of* $\mathbf{V}^T$

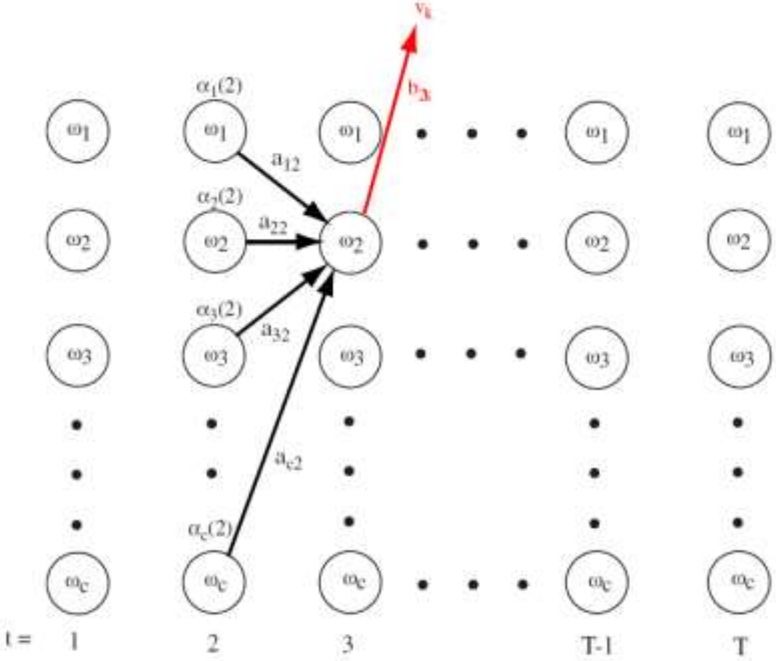
A trellis diagram (网格图)

$$\alpha_2(3) = \left[\sum_{i=1}^{c} \alpha_i(2) a_{i2}\right] b_{2k}$$

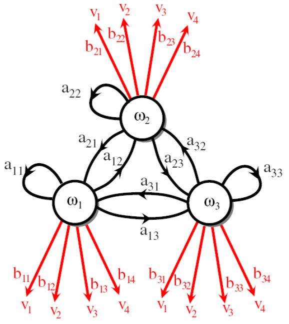
$t = 3$

$j = 2$

$v(t) = v_k$



The Evaluation Problem for HMM (Cont.)



An illustrative example

$\Omega = \{\omega_1, \omega_2, \ldots, \omega_4\}\ (c = 4) \qquad \mathcal{V} = \{v_1, v_2, \ldots, v_5\}\ (K = 5)$

$$\mathbf{A} = [a_{ij}]_{c\times c} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.2 & 0.3 & 0.1 & 0.4 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0.8 & 0.1 & 0.0 & 0.1 \end{pmatrix}$$

$$\mathbf{B} = [b_{jk}]_{c\times K} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0.4 & 0.1 & 0.2 \\ 0 & 0.1 & 0.1 & 0.7 & 0.1 \\ 0 & 0.5 & 0.2 & 0.1 & 0.2 \end{pmatrix}$$

$$\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_c) = \boxed{(0, 1, 0, 0)}$$

Specific properties for $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$

- □ ω_1 can be viewed as an ***absorbing*** state, which won't transit to other states once entered
- □ at state ω_1 , **only the symbol** v_1 **is emitted**
- □ at states other than ω_1 , **the symbol** v_1 **won't be emitted**
- □ the **initial state** should be ω_2

The Evaluation Problem for HMM (Cont.)

The forward procedure for evaluating $P(\mathbf{V}^5 \mid \boldsymbol{\theta})$ with $\mathbf{V}^5 = \{v_4, v_2, v_4, v_3, v_1\}$

	v_4	v_2	v_4	v_3	v_1
ω_1	0	0	0	0	.0011
ω_2	1	.09	.0052	.0024	0
ω_3	0	.01	.0077	.0002	0
ω_4	0	.20	.0057	.0007	0
$t =$	1	2	3	4	5

$$\mathbf{A} = [a_{ij}]_{c \times c} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.2 & 0.3 & 0.1 & 0.4 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0.8 & 0.1 & 0.0 & 0.1 \end{pmatrix}$$

$$\mathbf{B} = [b_{jk}]_{c \times K} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0.4 & 0.1 & 0.2 \\ 0 & 0.1 & 0.1 & 0.7 & 0.1 \\ 0 & 0.5 & 0.2 & 0.1 & 0.2 \end{pmatrix}$$

$$\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_c) = (0, 1, 0, 0)$$

The value of $\alpha_j(t)$ *is shown in the circles of the trellis*

$$\alpha_j(t) = \left[\sum_{i=1}^{c} \alpha_i(t-1) a_{ij}\right] b_{jv(t)}$$

The Evaluation Problem for HMM (Cont.)

HMM backward algorithm

$P(\mathbf{V}^T \mid \boldsymbol{\theta}) = \sum_{j=1}^{c} \pi_j b_{jv(1)} \beta_j(1)$

Let $\beta_j(t) = P(v(t+1), v(t+2), \ldots, v(T) \mid \omega(t) = \omega_j, \boldsymbol{\theta})$

the probability of observing the rest T - t symbols in $\mathbf{V}^T$ *given that the hidden state at step t is* ω_j

Then, $\beta_j(t)\ (1 \leq j \leq c, 1 \leq t \leq T)$ can be calculated recursively as:

$$\beta_j(T) = 1$$

$$\beta_j(t) = \sum_{i=1}^{c} P(v(t+1), \omega(t+1) = \omega_i, v(t+2), \ldots, v(T) \mid \omega(t) = \omega_j, \boldsymbol{\theta})$$

$$= \sum_{i=1}^{c} P(v(t+2), \ldots, v(T) \mid \omega(t+1) = \omega_i, \boldsymbol{\theta}) \cdot P(\omega_i \mid \omega_j, \boldsymbol{\theta}) \cdot P(v(t+1) \mid \omega_i, \boldsymbol{\theta})$$

$$= \sum_{i=1}^{c} \beta_i(t+1) a_{ji} b_{iv(t+1)}$$

The Evaluation Problem for HMM (Cont.)

$\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$: the complete set of HMM parameters

$\mathbf{V}^T$: the observed symbol sequence

to evaluate $P(\mathbf{V}^T \mid \boldsymbol{\theta})$

Pseudo-code for HMM backward algorithm

```
Initialize $t = T$ and $\beta_j(T) = 1 \ (1 \leq j \leq c)$
For $t = T - 1$ to $1$
    For $j = 1$ to $c$
        $\beta_j(t) = \sum_{i=1}^{c} \beta_i(t+1) a_{ji} b_{iv(t+1)}$
    End
End
Return $P(\mathbf{V}^T \mid \boldsymbol{\theta}) = \sum_{j=1}^{c} \pi_j b_{jv(1)} \beta_j(1)$
```

Computational complexity

$\mathcal{O}(c^T \cdot T)$

$\mathcal{O}(c^2 \cdot T)$

The Decoding Problem for HMM

$\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$: the HMM parameters

$\mathbf{V}^T$: the observed symbol sequence

to identify the state sequence

$$\boldsymbol{\omega}^* = \arg\max_{\boldsymbol{\omega}^T} P(\boldsymbol{\omega}^T \mid \mathbf{V}^T, \boldsymbol{\theta})$$

$$\boldsymbol{\omega}^* = \arg\max_{\boldsymbol{\omega}^T} P(\boldsymbol{\omega}^T \mid \mathbf{V}^T, \boldsymbol{\theta}) \Longleftrightarrow \boldsymbol{\omega}^* = \arg\max_{\boldsymbol{\omega}^T} P(\boldsymbol{\omega}^T, \mathbf{V}^T \mid \boldsymbol{\theta})$$

A straightforward decoding

$$\boldsymbol{\omega}^* = \arg\max_{\boldsymbol{\omega}^T} P(\boldsymbol{\omega}^T, \mathbf{V}^T \mid \boldsymbol{\theta})$$

$$P(\boldsymbol{\omega}^T \mid \boldsymbol{\theta}) = \prod_{t=1}^{T} P(\omega(t) \mid \omega(t-1)) \;\; (\textit{let } P(\omega(1) \mid \omega(0)) = \pi_{\omega(1)})$$

$$P(\mathbf{V}^T \mid \boldsymbol{\omega}^T, \boldsymbol{\theta}) = \prod_{t=1}^{T} b_{\omega(t)v(t)}$$

$$\boldsymbol{\omega}^* = \arg\max_{\boldsymbol{\omega}^T} \prod_{t=1}^{T} P(\omega(t) \mid \omega(t-1)) \cdot b_{\omega(t)v(t)}$$

Computational complexity: $\mathcal{O}(c^T \cdot T)$!

Infeasible!

The Decoding Problem for HMM (Cont.)

The Viterbi algorithm

Let $\delta_j(t) = \max_{\omega(1),\ldots,\omega(t-1)} P(\omega(1),\ldots,\omega(t-1),\omega(t)=\omega_j, v(1),\ldots,v(t) \mid \boldsymbol{\theta})$

the highest probability (best score) of the state sequence and observed symbols till step t, where the state at step t is ω_j

Similar to the forward and backward evaluation algorithm, $\delta_j(t)\ (1 \leq j \leq c, 1 \leq t \leq T)$ *can be calculated recursively based on dynamic programming (动态规划)*

Andrew J. Viterbi
Founder of Qualcomm
(1935-)

The Decoding Problem for HMM (Cont.)

$\delta_j(t)\ (1 \le j \le c, 1 \le t \le T)$ can be calculated recursively as:

$$\begin{aligned}\delta_j(1) = P(\omega(1) = \omega_j, v(1) \mid \boldsymbol{\theta}) &= P(\omega(1) = \omega_j \mid \boldsymbol{\theta}) \cdot P(v(1) \mid \omega_j, \boldsymbol{\theta}) \\ &= \pi_j b_{jv(1)}\end{aligned}$$

$$\begin{aligned}\delta_j(t) &= \max_{\omega(1),\ldots,\omega(t-1)} P(\omega(1), \ldots, \omega(t-1), \omega(t) = \omega_j, v(1), \ldots, v(t) \mid \boldsymbol{\theta}) \\ &= \max_{\omega(t-1)} \left[\max_{\omega(1),\ldots,\omega(t-2)} P(\omega(1), \ldots, \omega(t-1), \omega(t) = \omega_j, v(1), \ldots, v(t) \mid \boldsymbol{\theta}) \right] \\ &= \max_{1 \le i \le c} \left[\max_{\omega(1),\ldots,\omega(t-2)} P(\omega(1), \ldots, \omega(t-2), \omega(t-1) = \omega_i, v(1), \ldots, v(t-1) \mid \boldsymbol{\theta}) \right. \\ &\qquad \left. \cdot P(\omega_j \mid \omega_i, \boldsymbol{\theta}) \cdot P(v(t) \mid \omega_j, \boldsymbol{\theta}) \right] \\ &= \left[\max_{1 \le i \le c} \delta_i(t-1) a_{ij} \right] b_{jv(t)}\end{aligned}$$

The Decoding Problem for HMM (Cont.)

$\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$: the HMM parameters

$\mathbf{V}^T$: the observed symbol sequence

to identify the state sequence

$$\boldsymbol{\omega}^* = \arg\max_{\boldsymbol{\omega}^T} P(\boldsymbol{\omega}^T \mid \mathbf{V}^T, \boldsymbol{\theta})$$

Pseudo-code for the Viterbi algorithm

```
Initialize $\delta_j(1) = \pi_j b_{jv(1)}$ and $\psi_j(1) = 0 \ (1 \le j \le c)$
For $t = 2$ to $T$
    For $j = 1$ to $c$
        $\delta_j(t) = \left[\max_{1 \le i \le c} \delta_i(t-1) a_{ij}\right] b_{jv(t)};\ \psi_j(t) = \arg\max_{1 \le i \le c} \delta_i(t-1) a_{ij}$
    End
End
Decode $\omega^*(T) = \arg\max_{1 \le j \le c} \delta_j(T)$
Decode $\omega^*(t) = \psi_{\omega^*(t+1)}(t+1) \ (1 \le t \le T-1)$ with path backtracking (路径回溯)
```

Computational complexity

$\mathcal{O}(c^T \cdot T)$

$\mathcal{O}(c^2 \cdot T)$

The Learning Problem for HMM

$\mathbf{V}^T$: the observed symbol sequence → *to identify $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$ which which maximizes $P(\mathbf{V}^T \mid \boldsymbol{\theta})$*

*Generally, there is **no known algorithm** which can obtain the **optimal solution** to the above problem*

Try to find a **local optimum** based on iterative updating: in each iteration, update $\boldsymbol{\theta}$ to $\hat{\boldsymbol{\theta}}$ such that $P(\mathbf{V}^T \mid \hat{\boldsymbol{\theta}}) \geq P(\mathbf{V}^T \mid \boldsymbol{\theta})$

The Baum-Welch algorithm

*a.k.a. **forward-backward algorithm**, which is an instantiation of the famous **Expectation-Maximization (EM)** procedure*

Leonard E. Baum (1931-2017)

Lloyd R. Welch (1927-2024)

The Learning Problem for HMM (Cont.)

The Baum-Welch algorithm

Let $\gamma_{ij}(t) = P(\omega(t) = \omega_i, \omega(t+1) = \omega_j \mid \mathbf{V}^T, \boldsymbol{\theta})$

the probability of being in state ω_i *at step t, and state* ω_j *at step* t+1, given the observed symbol sequence

$$\gamma_{ij}(t) = P(\omega(t) = \omega_i, \omega(t+1) = \omega_j \mid \mathbf{V}^T, \boldsymbol{\theta})$$

$$= \frac{P(v(1), \ldots, v(t), \omega(t) = \omega_i, \omega(t+1) = \omega_j, v(t+1), \ldots, v(T) \mid \boldsymbol{\theta})}{P(\mathbf{V}^T \mid \boldsymbol{\theta})}$$

$$= \frac{\alpha_i(t)\, a_{ij}\, b_{jv(t+1)}\, \beta_j(t+1)}{P(\mathbf{V}^T \mid \boldsymbol{\theta})}$$

$$= \frac{\alpha_i(t)\, a_{ij}\, b_{jv(t+1)}\, \beta_j(t+1)}{\sum_{i=1}^{c} \sum_{j=1}^{c} \alpha_i(t)\, a_{ij}\, b_{jv(t+1)}\, \beta_j(t+1)}$$

The Learning Problem for HMM (Cont.)

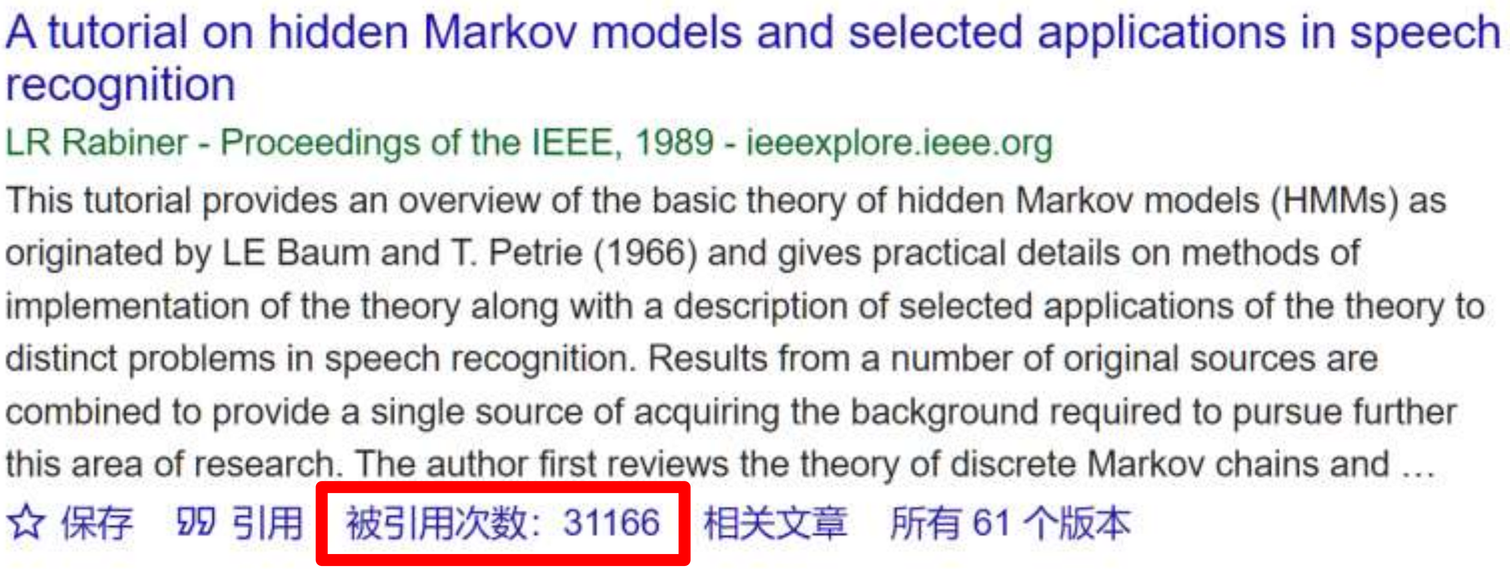
Pseudo-code for the Baum-Welch algorithm

```
1. Randomly initialize $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$
2. Repeat
3.     Estimate $\alpha_j(t)\ (1 \le j \le c, 1 \le t \le T)$ by invoking the forward algorithm
4.     Estimate $\beta_j(t)\ (1 \le j \le c, 1 \le t \le T)$ by invoking the backward algorithm
5.     Set $\gamma_{ij}(t) = \frac{\alpha_i(t)\, a_{ij}\, b_{jv(t+1)}\, \beta_j(t+1)}{\sum_{i=1}^{c} \sum_{j=1}^{c} \alpha_i(t)\, a_{ij}\, b_{jv(t+1)}\, \beta_j(t+1)}$ $(1 \le i, j \le c, 1 \le t \le T-1)$
6.     Set $\hat{\boldsymbol{\theta}} = \{\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\boldsymbol{\pi}}\}$ such that $\forall\ 1 \le i, j \le c, 1 \le k \le K:$
           $\hat{\pi}_i = \sum_{j=1}^{c} \gamma_{ij}(1)$   $\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_{ij}(t)}{\sum_{t=1}^{T-1} \sum_{j=1}^{c} \gamma_{ij}(t)}$   $\hat{b}_{ik} = \frac{\sum_{t=1, v(t)=v_k}^{T-1} \sum_{j=1}^{c} \gamma_{ij}(t)}{\sum_{t=1}^{T-1} \sum_{j=1}^{c} \gamma_{ij}(t)}$
7.     Update $\boldsymbol{\theta} \leftarrow \hat{\boldsymbol{\theta}}$
8. Until convergence
```

practical convergence condition: $\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\| \le \epsilon$

To have the full story on HMM……

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. ***Proceedings of the IEEE***, 1989, 77(2): 257-286



A tutorial on hidden Markov models and selected applications in speech recognition

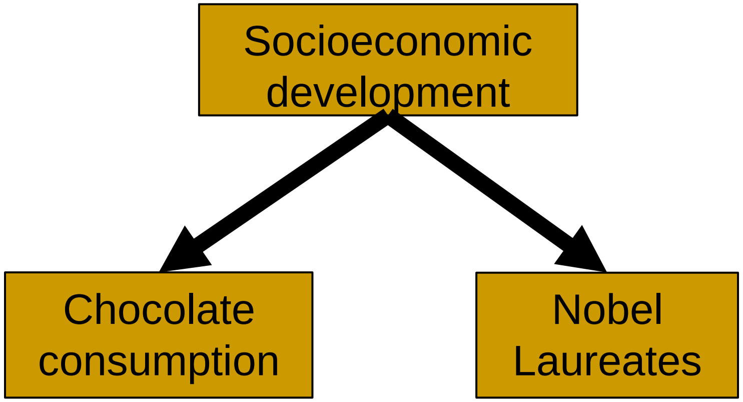
LR Rabiner - Proceedings of the IEEE, 1989 - ieeexplore.ieee.org

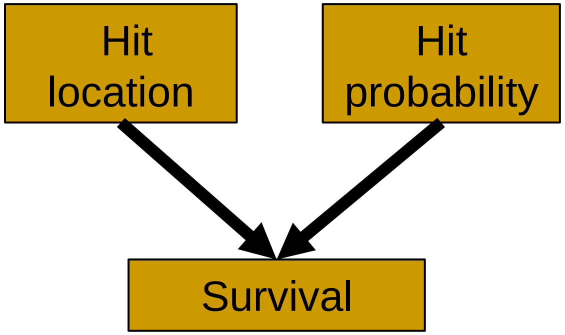
This tutorial provides an overview of the basic theory of hidden Markov models (HMMs) as originated by LE Baum and T. Petrie (1966) and gives practical details on methods of implementation of the theory along with a description of selected applications of the theory to distinct problems in speech recognition. Results from a number of original sources are combined to provide a single source of acquiring the background required to pursue further this area of research. The author first reviews the theory of discrete Markov chains and …

Related Topic II

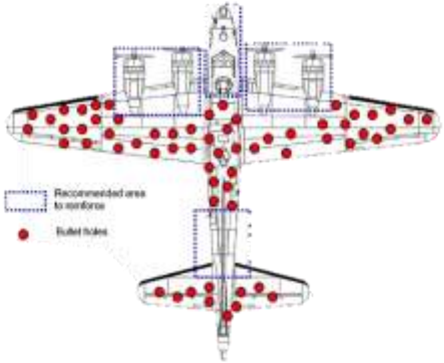
Bayesian Belief Network

Decision: a tale of two sides (Cont.)





- The first example is called "confounding bias"
- The second example is called "selection bias"



Directed Acyclic Graph (DAG; 有向无环图)

$G = (V, E)$

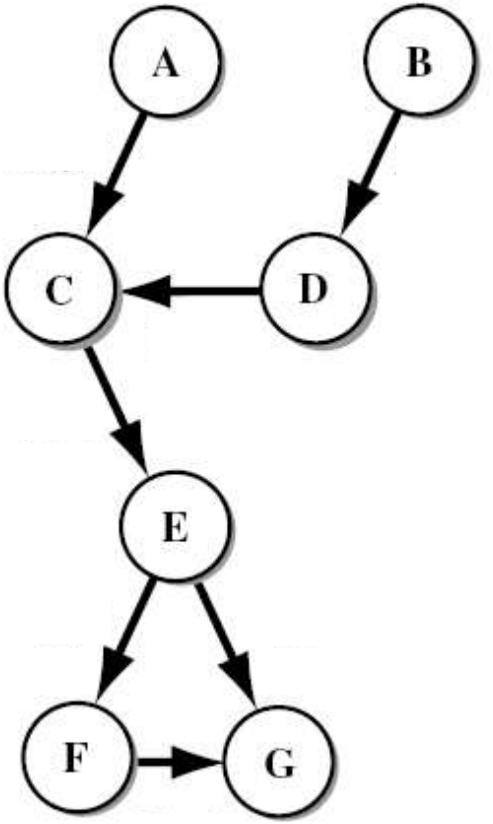
- □ V: a set of **nodes** in graph G
- □ E: a set of **directed edges** in G

Basic assumption: no directed loop in *G*

An illustrative example

$$V = \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}, \mathbf{F}, \mathbf{G}\} \quad (|V| = 7)$$

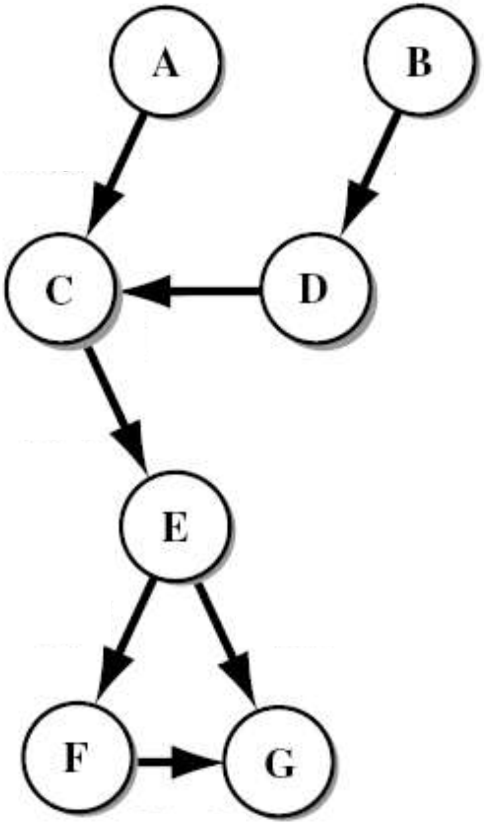
$$E = \{(\mathbf{A}, \mathbf{C}), (\mathbf{B}, \mathbf{D}), (\mathbf{D}, \mathbf{C}), (\mathbf{C}, \mathbf{E}), (\mathbf{E}, \mathbf{F}), (\mathbf{E}, \mathbf{G}), (\mathbf{F}, \mathbf{G})\} \quad (|E| = 7)$$



Bayesian Belief Network (贝叶斯置信网)

The goal of Bayesian belief network

Model the ***joint distribution of a set of random variables*** *w.r.t. the network's DAG structure*



Notations

- □ *Node*: $\mathbf{A}, \mathbf{B}, \ldots$
- □ *Random Variable*: $\mathbf{a}, \mathbf{b}, \ldots$
- □ *Values of Random Variable*: $\{a_1, a_2, \ldots\}, \ldots$
- □ *Parent variables*: $\mathcal{G}(\mathbf{a}), \mathcal{G}(\mathbf{b}), \ldots$

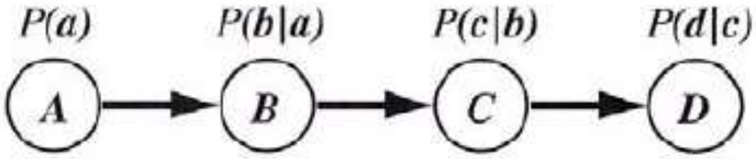
  e.g. $\mathcal{G}(\mathbf{c}) = \{\mathbf{a}, \mathbf{d}\}, \mathcal{G}(\mathbf{f}) = \{\mathbf{e}\}$

joint distribution w.r.t. the DAG

The joint distribution can ***be factorized into the product of the conditional probability*** *of each random variable given its parent variables*

Bayesian Belief Network (Cont.)

DAG Example I



$$P(\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}) = P(\mathbf{a}\mid\mathcal{G}(\mathbf{a}))\cdot P(\mathbf{b}\mid\mathcal{G}(\mathbf{b}))\cdot P(\mathbf{c}\mid\mathcal{G}(\mathbf{c}))\cdot P(\mathbf{d}\mid\mathcal{G}(\mathbf{d}))$$

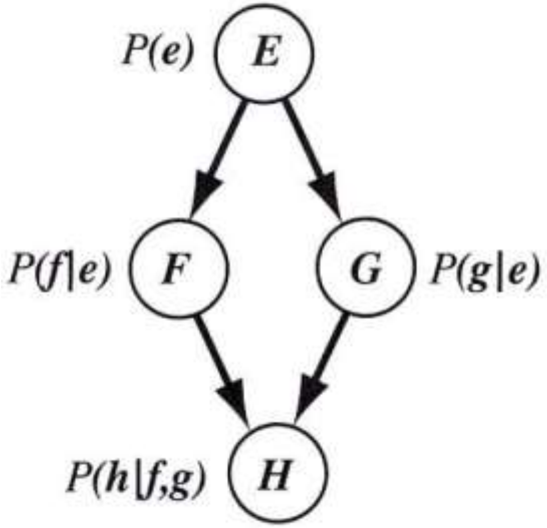
$$= P(\mathbf{a})\cdot P(\mathbf{b}\mid\mathbf{a})\cdot P(\mathbf{c}\mid\mathbf{b})\cdot P(\mathbf{d}\mid\mathbf{c})$$

$$P(\mathbf{d}) = \sum_{\mathbf{a},\mathbf{b},\mathbf{c}} P(\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}) = \sum_{\mathbf{a},\mathbf{b},\mathbf{c}} P(\mathbf{a})\cdot P(\mathbf{b}\mid\mathbf{a})\cdot P(\mathbf{c}\mid\mathbf{b})\cdot P(\mathbf{d}\mid\mathbf{c})$$

$$= \underbrace{\sum_{\mathbf{c}} P(\mathbf{d}\mid\mathbf{c}) \underbrace{\sum_{\mathbf{b}} P(\mathbf{c}\mid\mathbf{b}) \underbrace{\sum_{\mathbf{a}} P(\mathbf{b}\mid\mathbf{a})P(\mathbf{a})}_{P(\mathbf{b})}}_{P(\mathbf{c})}}_{P(\mathbf{d})}$$

Bayesian Belief Network (Cont.)

DAG Example II



$$P(\mathbf{e},\mathbf{f},\mathbf{g},\mathbf{h})$$

$$= P(\mathbf{e} \mid \mathcal{G}(\mathbf{e})) \cdot P(\mathbf{f} \mid \mathcal{G}(\mathbf{f})) \cdot P(\mathbf{g} \mid \mathcal{G}(\mathbf{g})) \cdot P(\mathbf{h} \mid \mathcal{G}(\mathbf{h}))$$

$$= P(\mathbf{e}) \cdot P(\mathbf{f} \mid \mathbf{e}) \cdot P(\mathbf{g} \mid \mathbf{e}) \cdot P(\mathbf{h} \mid \mathbf{f},\mathbf{g})$$

$$P(\mathbf{f},\mathbf{g},\mathbf{h}) = \sum_{\mathbf{e}} P(\mathbf{e},\mathbf{f},\mathbf{g},\mathbf{h})$$

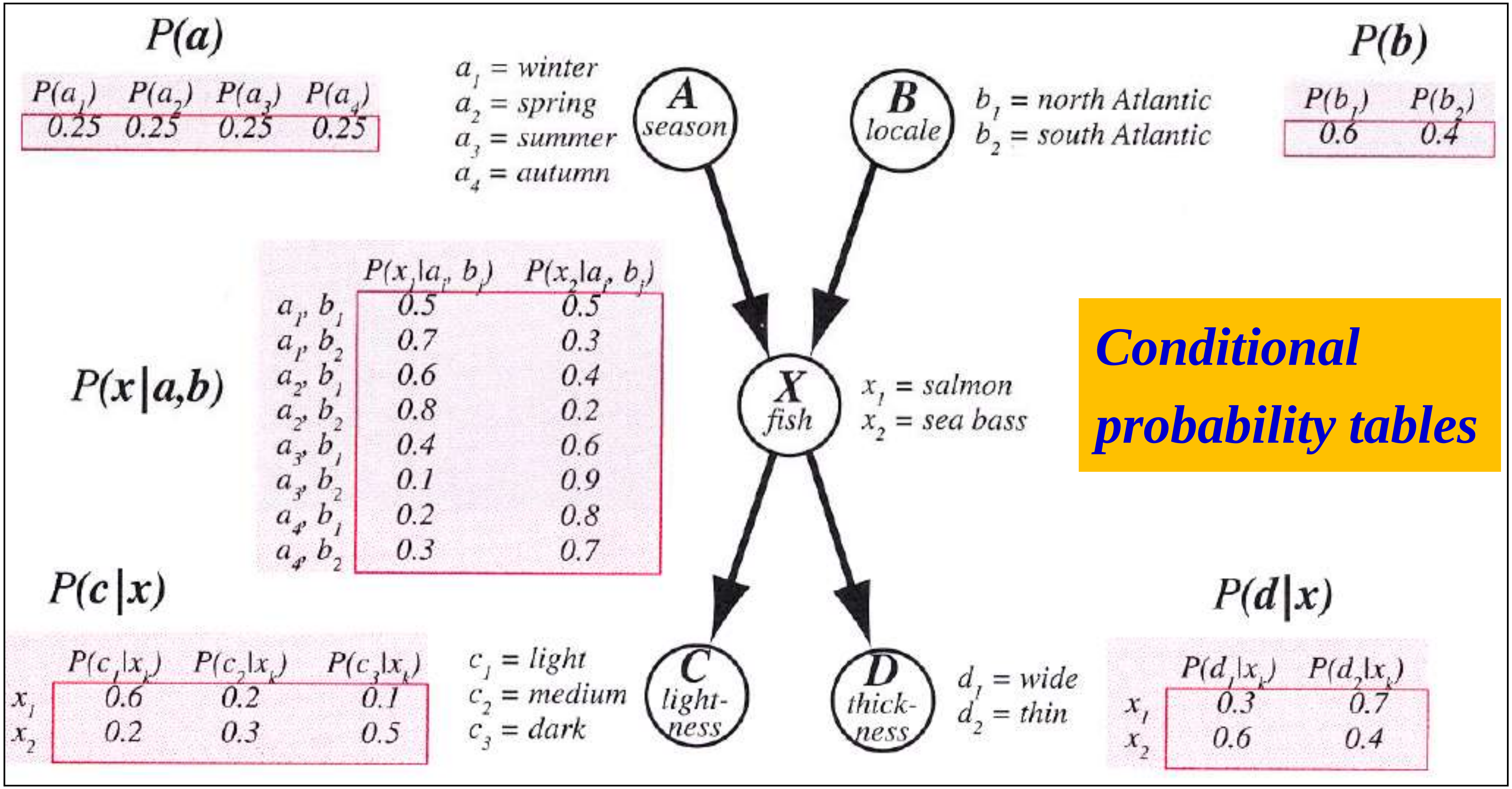
$$= P(\mathbf{h} \mid \mathbf{f},\mathbf{g}) \sum_{\mathbf{e}} P(\mathbf{e}) \cdot P(\mathbf{f} \mid \mathbf{e}) \cdot P(\mathbf{g} \mid \mathbf{e})$$

$$P(\mathbf{h}) = \sum_{\mathbf{e},\mathbf{f},\mathbf{g}} P(\mathbf{e},\mathbf{f},\mathbf{g},\mathbf{h})$$

$$= \sum_{\mathbf{e}} P(\mathbf{e}) \sum_{\mathbf{f},\mathbf{g}} P(\mathbf{f} \mid \mathbf{e}) \cdot P(\mathbf{g} \mid \mathbf{e}) \cdot P(\mathbf{h} \mid \mathbf{f},\mathbf{g})$$

Bayesian Belief Network (Cont.)

DAG Example III **Bayesian network for fish**



$P(a)$

$P(a_1)$	$P(a_2)$	$P(a_3)$	$P(a_4)$
0.25	0.25	0.25	0.25

a_1 = winter
a_2 = spring
a_3 = summer
a_4 = autumn

b_1 = north Atlantic
b_2 = south Atlantic

$P(b)$

$P(b_1)$	$P(b_2)$
0.6	0.4

$P(x|a,b)$

	$P(x_1\|a_i, b_j)$	$P(x_2\|a_i, b_j)$
a_1, b_1	0.5	0.5
a_1, b_2	0.7	0.3
a_2, b_1	0.6	0.4
a_2, b_2	0.8	0.2
a_3, b_1	0.4	0.6
a_3, b_2	0.1	0.9
a_4, b_1	0.2	0.8
a_4, b_2	0.3	0.7

x_1 = salmon
x_2 = sea bass

Conditional probability tables

$P(c|x)$

	$P(c_1\|x_k)$	$P(c_2\|x_k)$	$P(c_3\|x_k)$
x_1	0.6	0.2	0.1
x_2	0.2	0.3	0.5

c_1 = light
c_2 = medium
c_3 = dark

d_1 = wide
d_2 = thin

$P(d|x)$

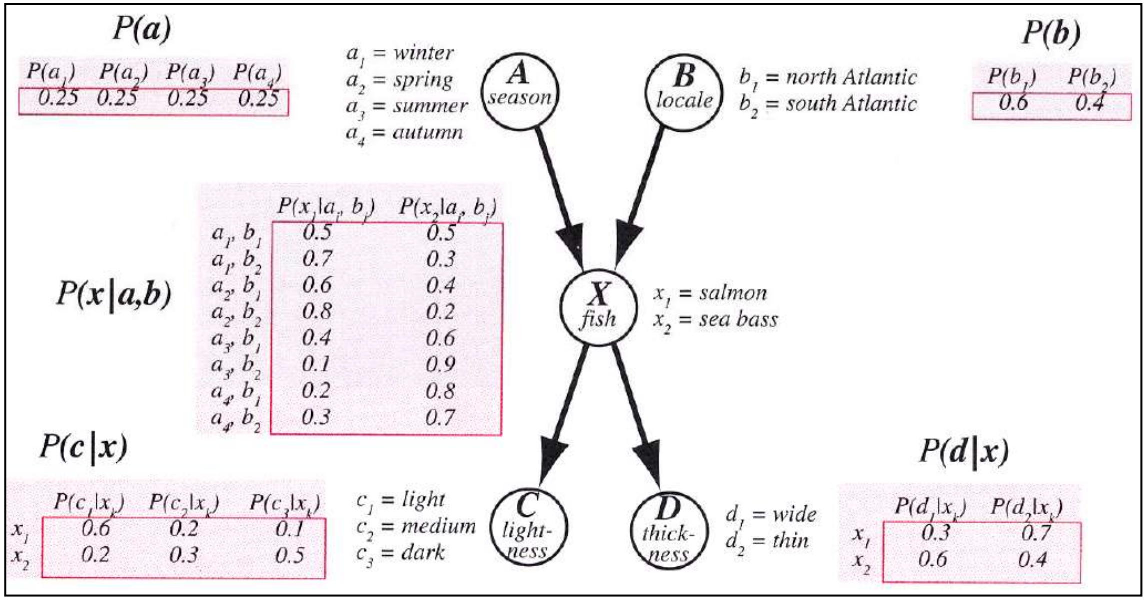
	$P(d_1\|x_k)$	$P(d_2\|x_k)$
x_1	0.3	0.7
x_2	0.6	0.4

Bayesian Belief Network (Cont.)

What is the probability *that the fish was caught in the* ***summer*** *in the* ***north Atlantic*** *and is* ***sea bass*** *that is* ***dark*** *and* ***thin****?*

$\mathbf{a} = a_3 \quad \mathbf{b} = b_1$

$\mathbf{x} = x_2 \quad \mathbf{c} = c_3 \quad \mathbf{d} = d_2$



a_1 = winter
a_2 = spring
a_3 = summer
a_4 = autumn

b_1 = north Atlantic
b_2 = south Atlantic

x_1 = salmon
x_2 = sea bass

c_1 = light
c_2 = medium
c_3 = dark

d_1 = wide
d_2 = thin

$P(\boldsymbol{a})$

$P(a_1)$	$P(a_2)$	$P(a_3)$	$P(a_4)$
0.25	0.25	0.25	0.25

$P(\boldsymbol{b})$

$P(b_1)$	$P(b_2)$
0.6	0.4

$P(\boldsymbol{x}|\boldsymbol{a},\boldsymbol{b})$

	$P(x_1\|a_i, b_j)$	$P(x_2\|a_i, b_j)$
a_1, b_1	0.5	0.5
a_1, b_2	0.7	0.3
a_2, b_1	0.6	0.4
a_2, b_2	0.8	0.2
a_3, b_1	0.4	0.6
a_3, b_2	0.1	0.9
a_4, b_1	0.2	0.8
a_4, b_2	0.3	0.7

$P(\boldsymbol{c}|\boldsymbol{x})$

	$P(c_1\|x_k)$	$P(c_2\|x_k)$	$P(c_3\|x_k)$
x_1	0.6	0.2	0.1
x_2	0.2	0.3	0.5

$P(\boldsymbol{d}|\boldsymbol{x})$

	$P(d_1\|x_k)$	$P(d_2\|x_k)$
x_1	0.3	0.7
x_2	0.6	0.4

$$P(a_3, b_1, x_2, c_3, d_2) = P(a_3) \cdot P(b_1) \cdot P(x_2 \mid a_3, b_1) \cdot P(c_3 \mid x_2) \cdot P(d_2 \mid x_2)$$

$$= 0.25 \times 0.6 \times 0.6 \times 0.5 \times 0.4$$

$$= 0.018$$

Bayesian Belief Network (Cont.)

*Suppose we know a fish is **light** and caught in the **south Atlantic**, **how shall we classify the fish**?*

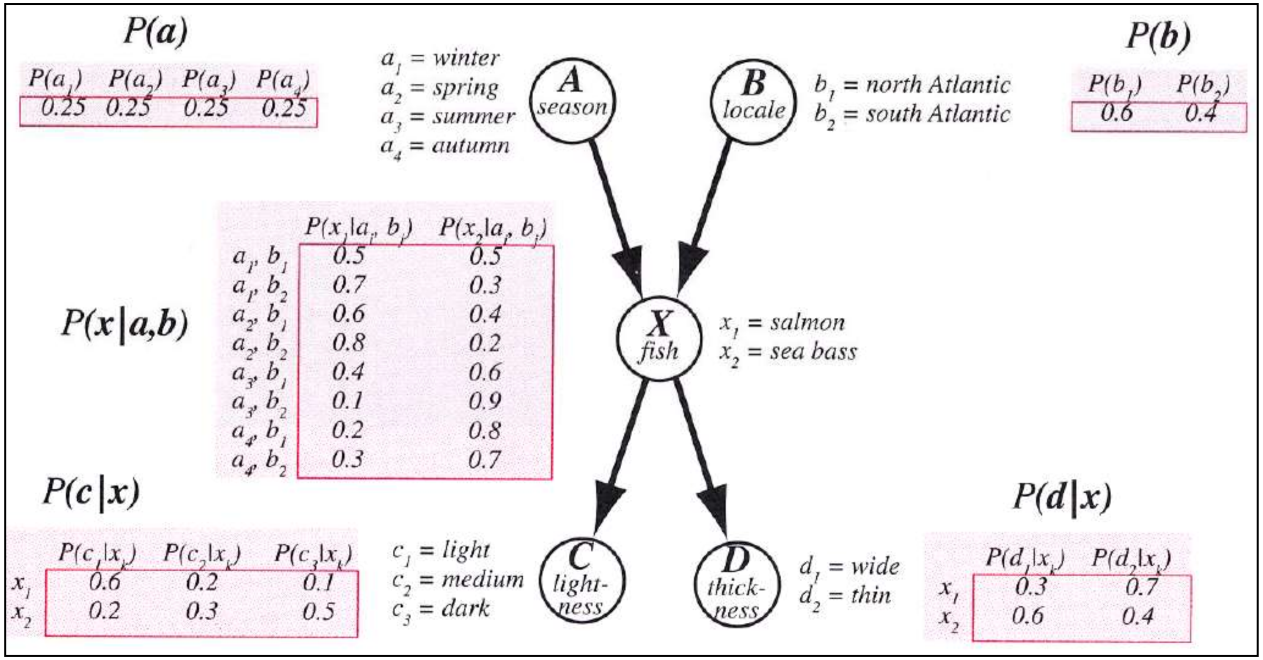
$\mathbf{b} = b_2 \quad \mathbf{c} = c_1$

evidence

$$P(\mathbf{x} = x_1 \mid b_2, c_1)$$

vs

$$P(\mathbf{x} = x_2 \mid b_2, c_1)$$



$P(a)$

$P(a_1)$	$P(a_2)$	$P(a_3)$	$P(a_4)$
0.25	0.25	0.25	0.25

a_1 = winter
a_2 = spring
a_3 = summer
a_4 = autumn

b_1 = north Atlantic
b_2 = south Atlantic

$P(b)$

$P(b_1)$	$P(b_2)$
0.6	0.4

$P(x|a,b)$

	$P(x_1\|a_i, b_j)$	$P(x_2\|a_i, b_j)$
a_1, b_1	0.5	0.5
a_1, b_2	0.7	0.3
a_2, b_1	0.6	0.4
a_2, b_2	0.8	0.2
a_3, b_1	0.4	0.6
a_3, b_2	0.1	0.9
a_4, b_1	0.2	0.8
a_4, b_2	0.3	0.7

x_1 = salmon
x_2 = sea bass

$P(c|x)$

	$P(c_1\|x_k)$	$P(c_2\|x_k)$	$P(c_3\|x_k)$
x_1	0.6	0.2	0.1
x_2	0.2	0.3	0.5

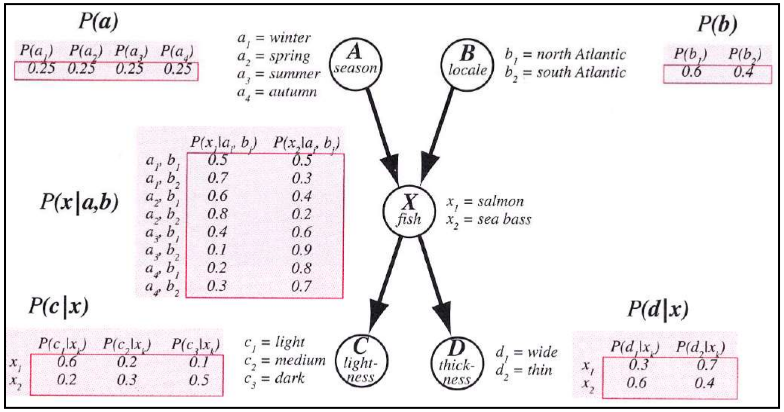
c_1 = light
c_2 = medium
c_3 = dark

d_1 = wide
d_2 = thin

$P(d|x)$

	$P(d_1\|x_k)$	$P(d_2\|x_k)$
x_1	0.3	0.7
x_2	0.6	0.4

Bayesian Belief Network (Cont.)



$P(a)$

$P(a_1)$	$P(a_2)$	$P(a_3)$	$P(a_4)$
0.25	0.25	0.25	0.25

a_1 = winter
a_2 = spring
a_3 = summer
a_4 = autumn

b_1 = north Atlantic
b_2 = south Atlantic

$P(b)$

$P(b_1)$	$P(b_2)$
0.6	0.4

$P(x|a,b)$

	$P(x_1\|a_i, b_j)$	$P(x_2\|a_i, b_j)$
a_1, b_1	0.5	0.5
a_1, b_2	0.7	0.3
a_2, b_1	0.6	0.4
a_2, b_2	0.8	0.2
a_3, b_1	0.4	0.6
a_3, b_2	0.1	0.9
a_4, b_1	0.2	0.8
a_4, b_2	0.3	0.7

x_1 = salmon
x_2 = sea bass

$P(c|x)$

	$P(c_1\|x_k)$	$P(c_2\|x_k)$	$P(c_3\|x_k)$
x_1	0.6	0.2	0.1
x_2	0.2	0.3	0.5

c_1 = light
c_2 = medium
c_3 = dark

d_1 = wide
d_2 = thin

$P(d|x)$

	$P(d_1\|x_k)$	$P(d_2\|x_k)$
x_1	0.3	0.7
x_2	0.6	0.4

$$P(x_1 \mid b_2, c_1)$$

$$= P(x_1, b_2, c_1)/P(b_2, c_1)$$

$$= \alpha \sum\nolimits_{\mathbf{a},\mathbf{d}} P(\mathbf{a}, x_1, b_2, c_1, \mathbf{d})$$

$$= \alpha \sum\nolimits_{\mathbf{a},\mathbf{d}} P(\mathbf{a}) P(b_2) P(x_1 \mid \mathbf{a}, b_2) P(c_1 \mid x_1) P(\mathbf{d} \mid x_1)$$

$$= \alpha P(b_2) P(c_1 \mid x_1) \left[\sum\nolimits_{\mathbf{a}} P(\mathbf{a}) P(x_1 \mid \mathbf{a}, b_2)\right] \left[\sum\nolimits_{\mathbf{d}} P(\mathbf{d} \mid x_1)\right]$$

$$= \alpha (0.4)(0.6)[(0.25)(0.7) + (0.25)(0.8) + (0.25)(0.1) + (0.25)(0.3)](1.0)$$

$$= \alpha 0.114$$

Similarly, we can have $P(x_2 \mid b_2, c_1) = \alpha 0.042$

$$P(x_1|b_2, c_1) = 0.73 \quad P(x_2|b_2, c_1) = 0.27$$

Bayesian Belief Network (Cont.)

Further Example **Bayesian network for xxx**

Summary

- Key issue for PR
  - Estimate prior and class-conditional pdf from training set
  - Basic assumption on training examples: *i.i.d.*
- Two strategies to the key issue
  - Parametric form for class-conditional pdf
    - Maximum likelihood (ML) estimation
    - Bayesian estimation
  - No parametric form for class-conditional pdf

Summary (Cont.)

- Maximum likelihood estimation
  - Settings: parameters as fixed but unknown values
  - The objective function: Log-likelihood function
  - Necessary conditions for ML estimation: gradient for the objective function should be zero vector
  - The Gaussian case
    - Unknown μ
    - Unknown μ and Σ

Summary (Cont.)

- Bayesian estimation
  - Settings: parameters as random variables
  - The general procedure
    - Phase I: *prior pdf* ➔ *posterior pdf* (for $\boldsymbol{\theta}$)
    - Phase II: *posterior pdf* (for $\boldsymbol{\theta}$) ➔ *class-conditional pdf* (for $\mathbf{x}$)
    - Phase III: *prediction* (Eq.22 [pp.91])
  - The Gaussian case
    - Unknown ˈ : univariate and multivariate

Summary (Cont.)

- Hidden Markov Model (HMM)
  - Parameters in HMM: $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$
  - Observed symbol sequence: $\mathbf{V}^T$
  - Three central problems in HMM
    - **Evaluation:** $P(\mathbf{V}^T \mid \boldsymbol{\theta})$, *the forward/backward algorithm*
    - **Decoding:** $\arg\max_{\boldsymbol{\omega}^T} P(\boldsymbol{\omega}^T \mid \mathbf{V}^T, \boldsymbol{\theta})$, *the Viterbi algorithm*
    - **Learning:** $\arg\max_{\boldsymbol{\theta}} P(\mathbf{V}^T \mid \boldsymbol{\theta})$, *the Baum-Welch algorithm*
- Bayesian Belief Network
  - The **DAG structure** for modeling joint distribution
  - Conditional probability tables