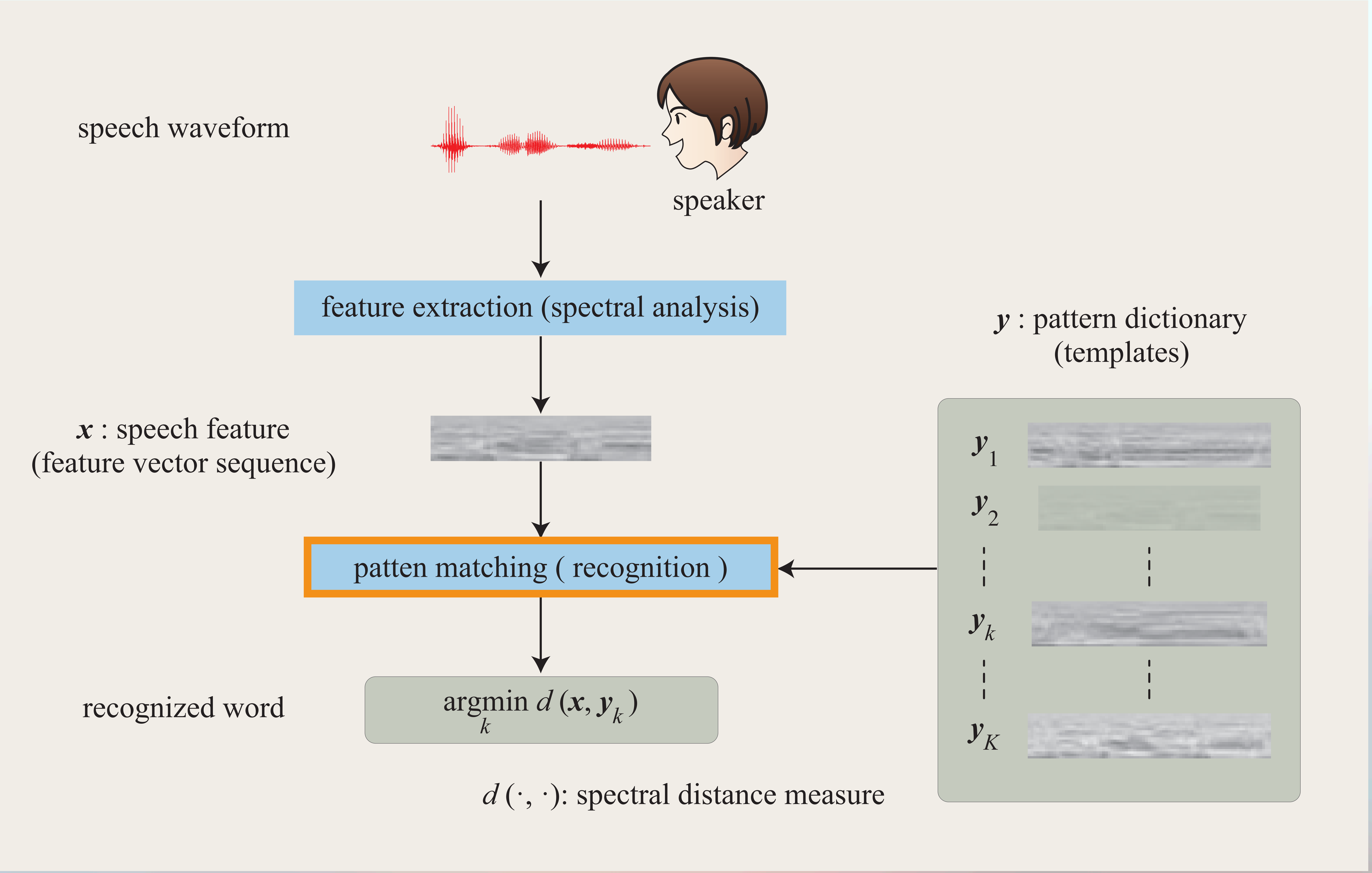
メディア情報学実験
メディア情報検索・認識実験（音声）

HMM算法

Hidden Markov Model Algorithms

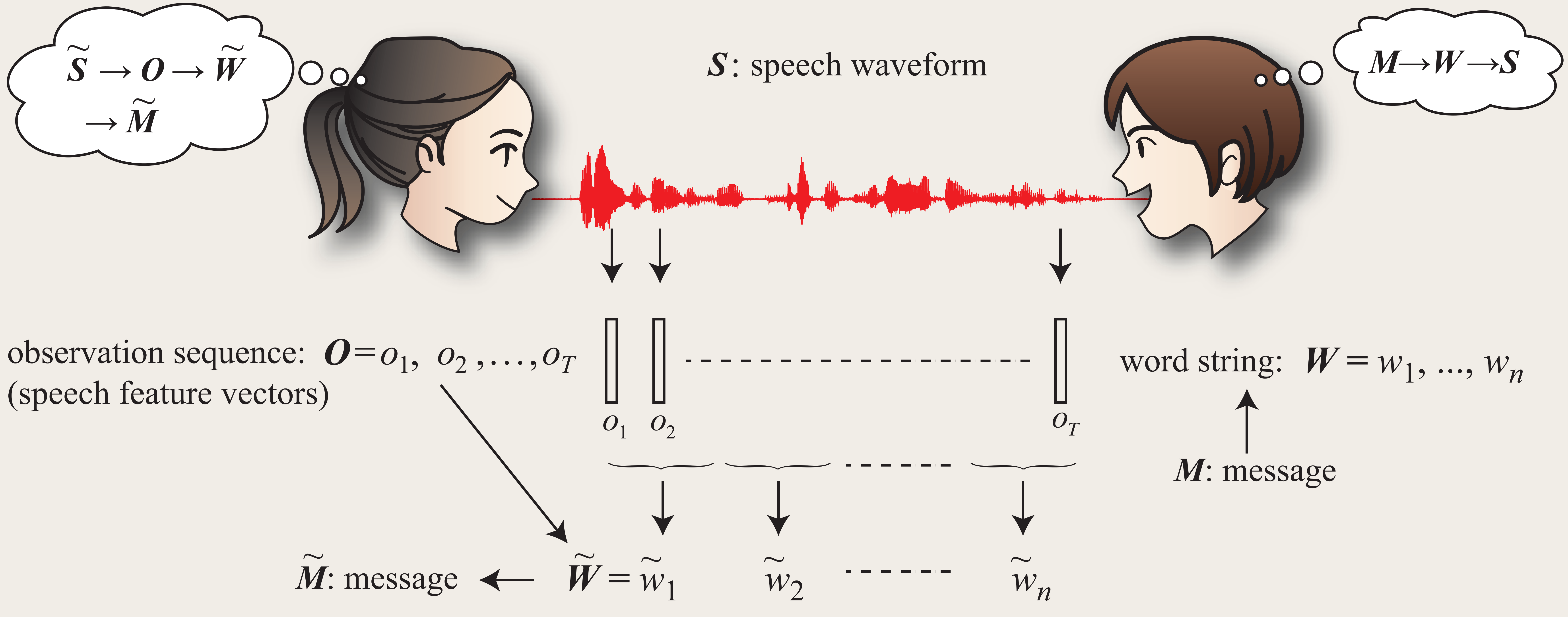
出席カードに学籍番号と氏名を記入し、2限の時間中に前方の箱に提出してください。



古典的な音声認識の pattern matchingでは「統計的音声認識」の枠組みにおいて隠れマルコフモデル（Hidden Markov Model: HMM）を利用する。

統計的音声認識

STATISTICAL SPEECH RECOGNITION



acoustic model　　language model

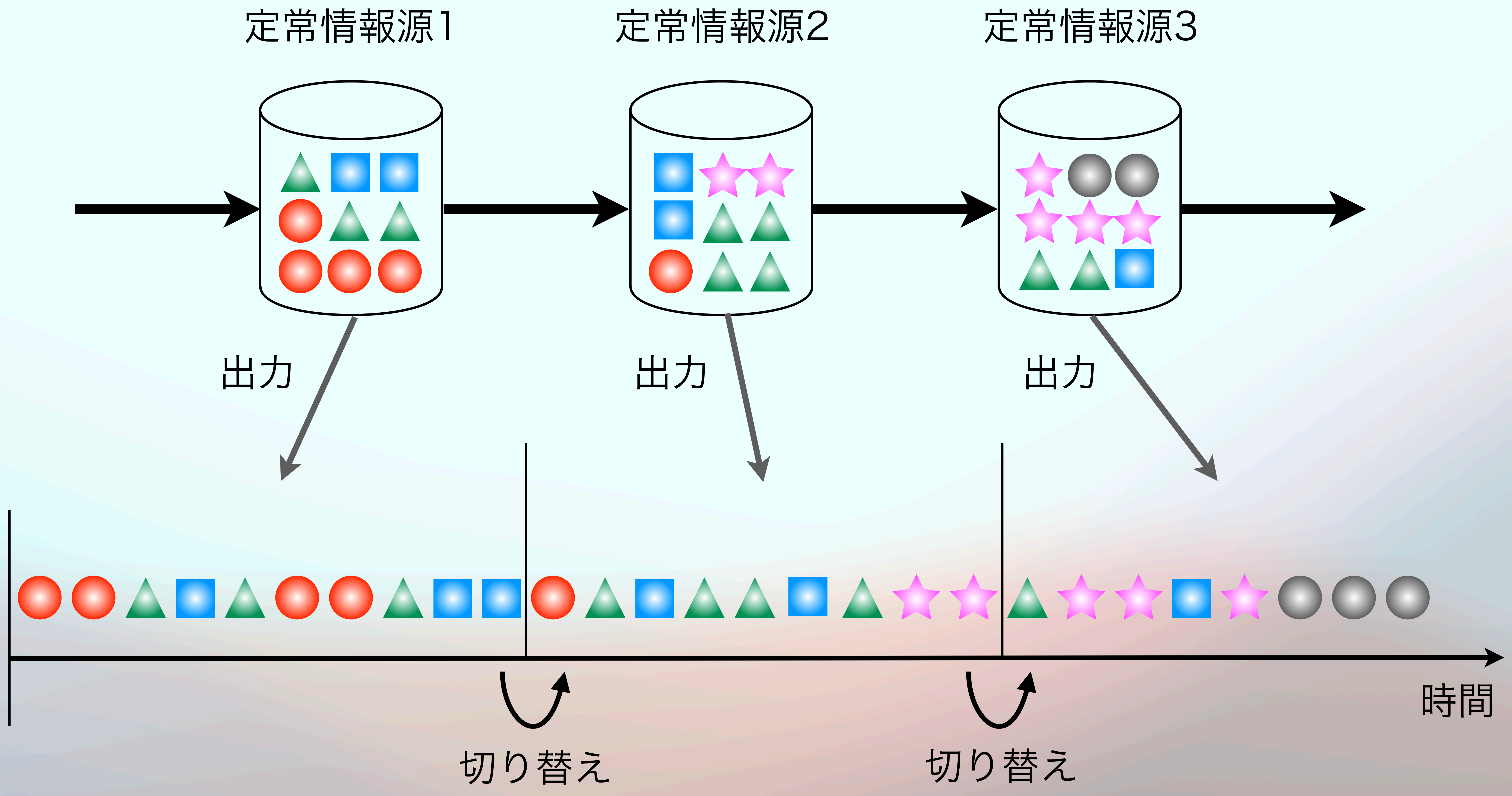
conditional probability of W given O : $P(W|O) = \dfrac{P(O|W)\,P(W)}{P(O)}$

recognized word string : $\tilde{W} = \underset{W}{\operatorname{argmax}}\, P(W|O)$

音声認識とは音声信号 S から単語列 W を推定すること。
特徴量時系列 O が与えられた条件で単語列 W が生成する条件付き確率を評価。

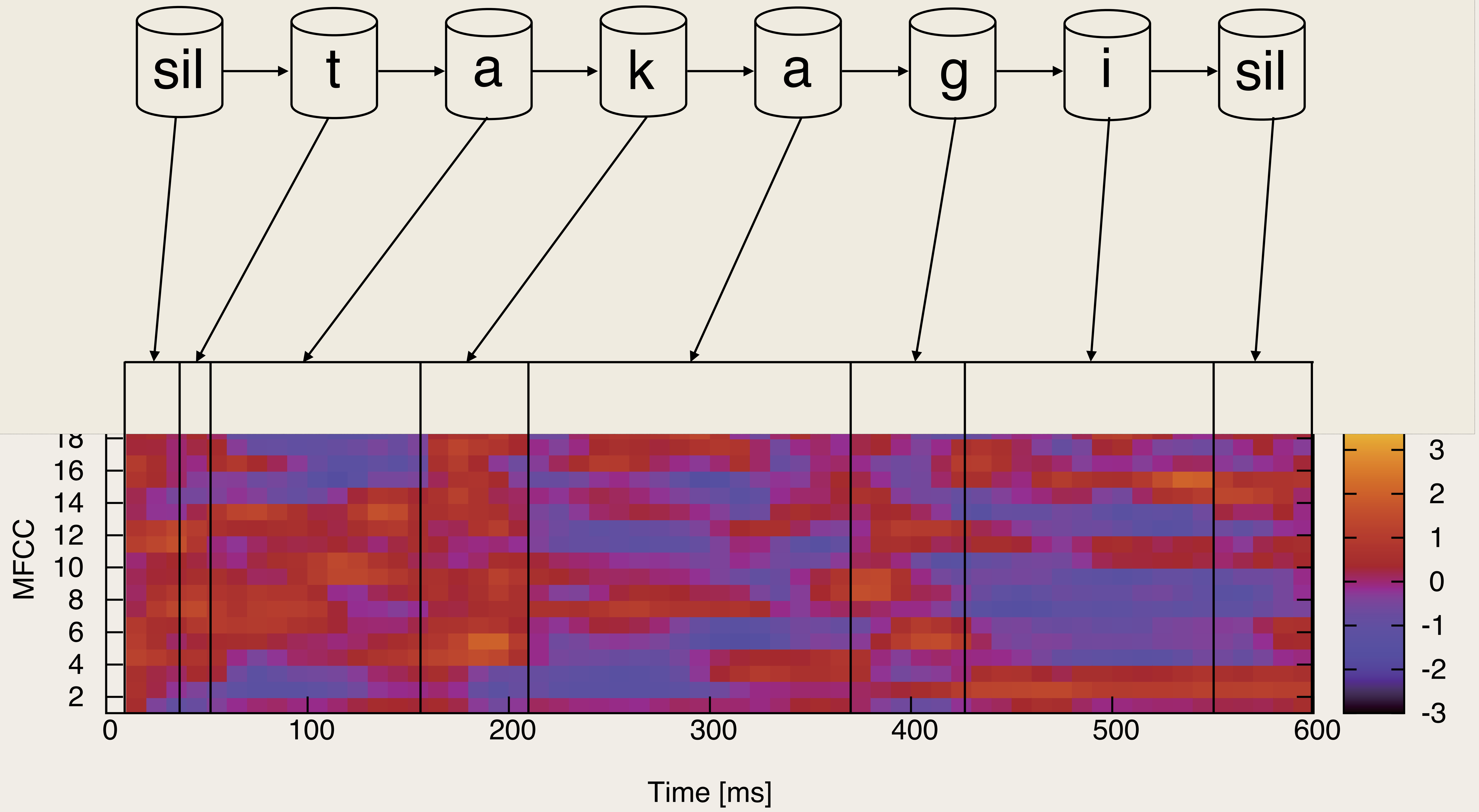
非定常信号源のモデル

異なる定常情報源を次々に切り替える



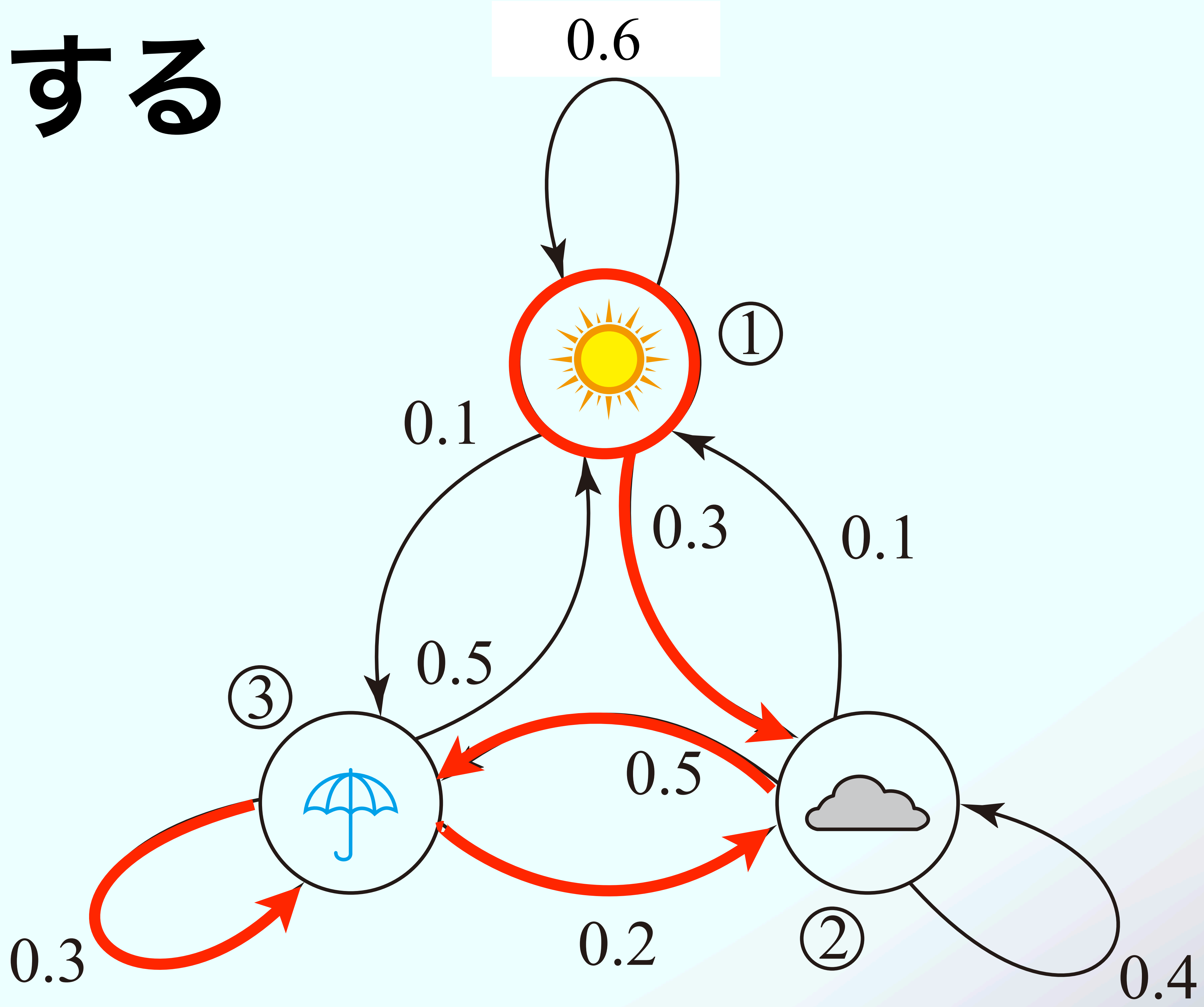
音素（定常情報源）が発音順に並び（切り替わり）言語音となる。

音声のスペクトル分布の変化のモデル

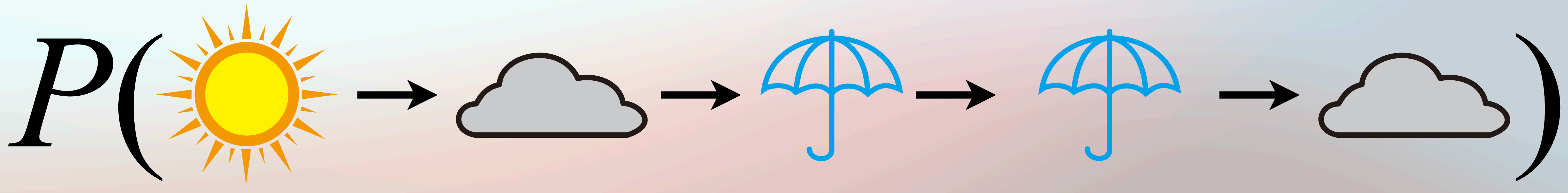


定常情報源は一定の分布の多次元実数ベクトル。単語の前後の sil は無音声。

天気変化をモデル化する



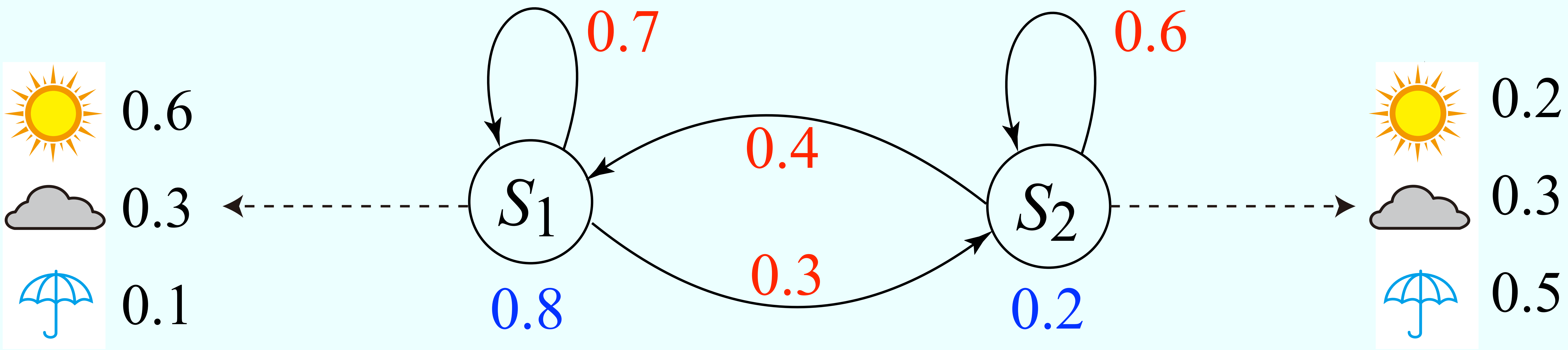
単純（一重）マルコフモデル



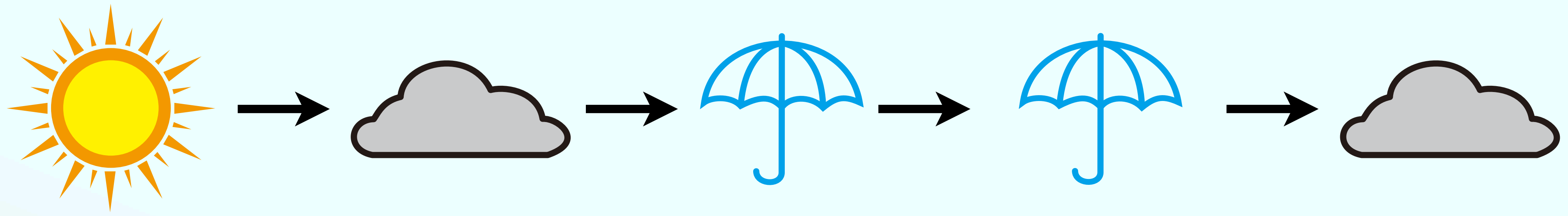
$$= 0.3 \times 0.5 \times 0.3 \times 0.2$$

$$= 0.009$$

天気の確率を計算：遷移確率の積



天気の隠れマルコフモデル：状態と天気は確率的に対応。

$P($  $)$

$$= 0.8\times0.6\times0.7\times0.3\times0.3\times0.5\times0.6\times0.5\times0.4\times0.3$$

$S_1 \quad S_1 \quad S_2 \quad S_2 \quad S_1$

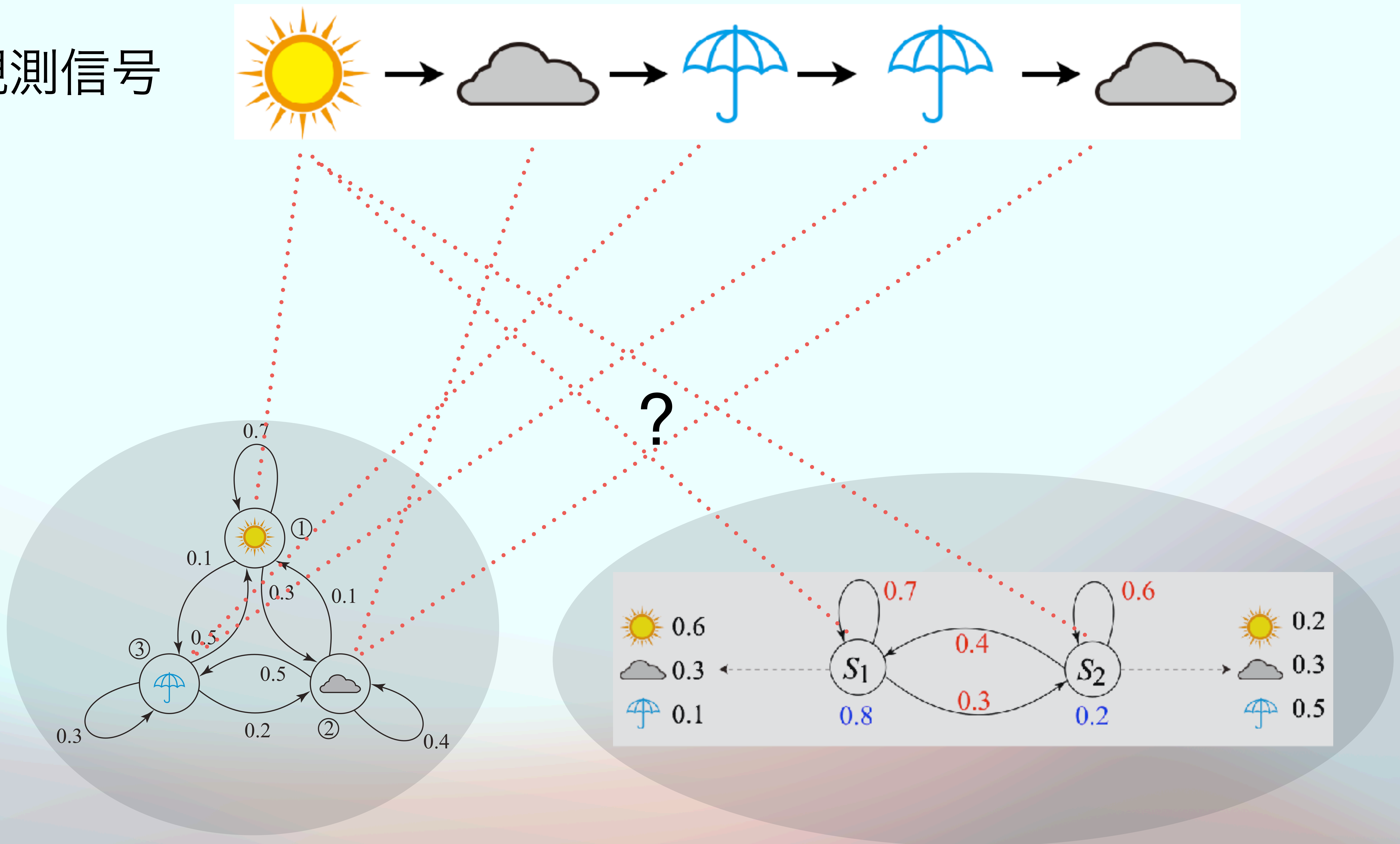
$$+0.2\times0.2\times0.4\times0.3\times0.3\times0.5\times0.4\times0.1\times0.3\times0.3$$

$S_2 \quad S_1 \quad S_2 \quad S_1 \quad S_2$

$+\ldots$

状態遷移は複数の可能性があり一意に定まらない

観測信号



マルコフモデル

観測信号から内部状態が一意に定まる

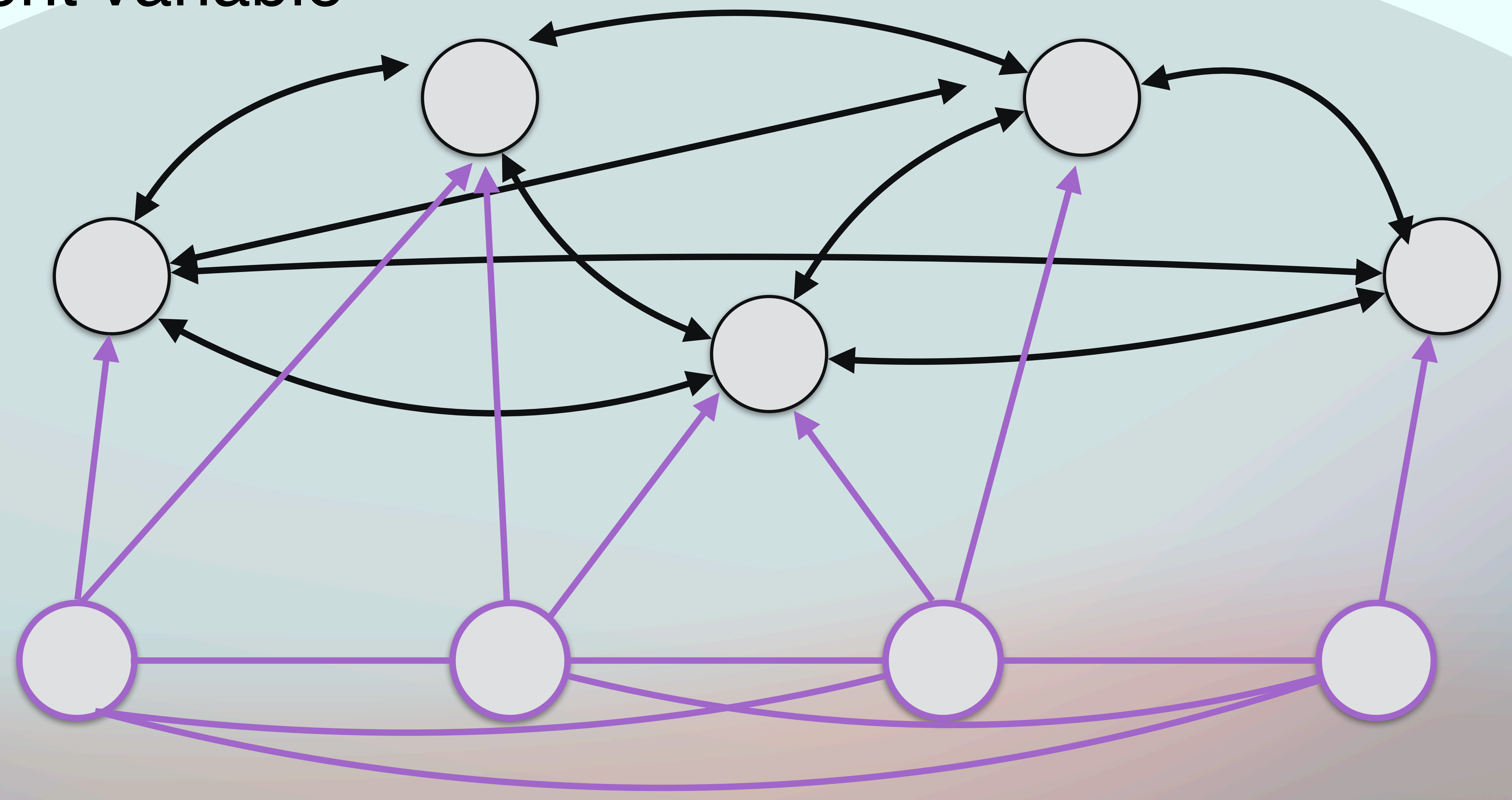
隠れマルコフモデル

観測信号から内部状態が一意に定まらない（隠れている）

観測信号

隠れ変数 hidden variable

潜在変数 latent variable



現象の背後にある要因を仮定し、それを隠れ／潜在変数として表す。

e.g. 別の地方の天気、海水温、地球規模の大気の状態

HMMの定式化

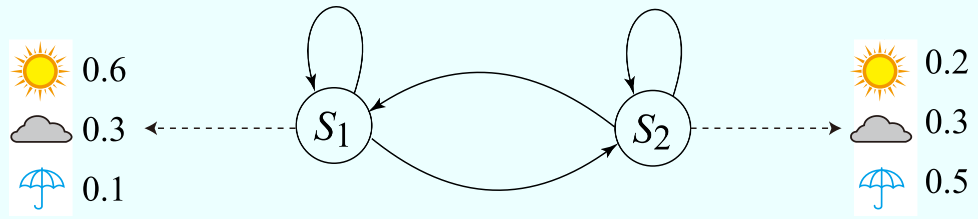
N：モデル λ の状態数

M：出力シンボル種類数

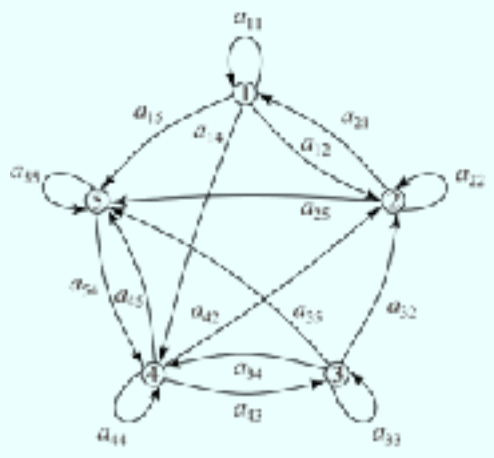
$\boldsymbol{A}$：状態遷移確率

$\boldsymbol{B}$：シンボル出力確率

$\boldsymbol{\pi}$：初期状態確率



N=2



N=5

☀ ☁ ☂ M=3

HMM $\lambda = (N, M, A, B, \pi)$

$\lambda = (A, B, \pi)$

0.2
0.3
0.5

$$\boldsymbol{\pi} = \{\pi i\} = (0.8\ 0.2)$$

$$\boldsymbol{A} = \{aij\} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix}$$

$$\boldsymbol{B} = \{bi(k)\} = \begin{pmatrix} b_1(☀) & b_1(☁) & b_1(☂) \\ b_2(☀) & b_2(☁) & b_2(☂) \end{pmatrix} = \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.2 & 0.3 & 0.5 \end{pmatrix}$$

HMMを使ってパターン認識

認識　観測系列 $\boldsymbol{O} = \boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T$ とモデル λ が与えられたとき

$P(\boldsymbol{O} \mid \lambda)$ を効率的に計算する

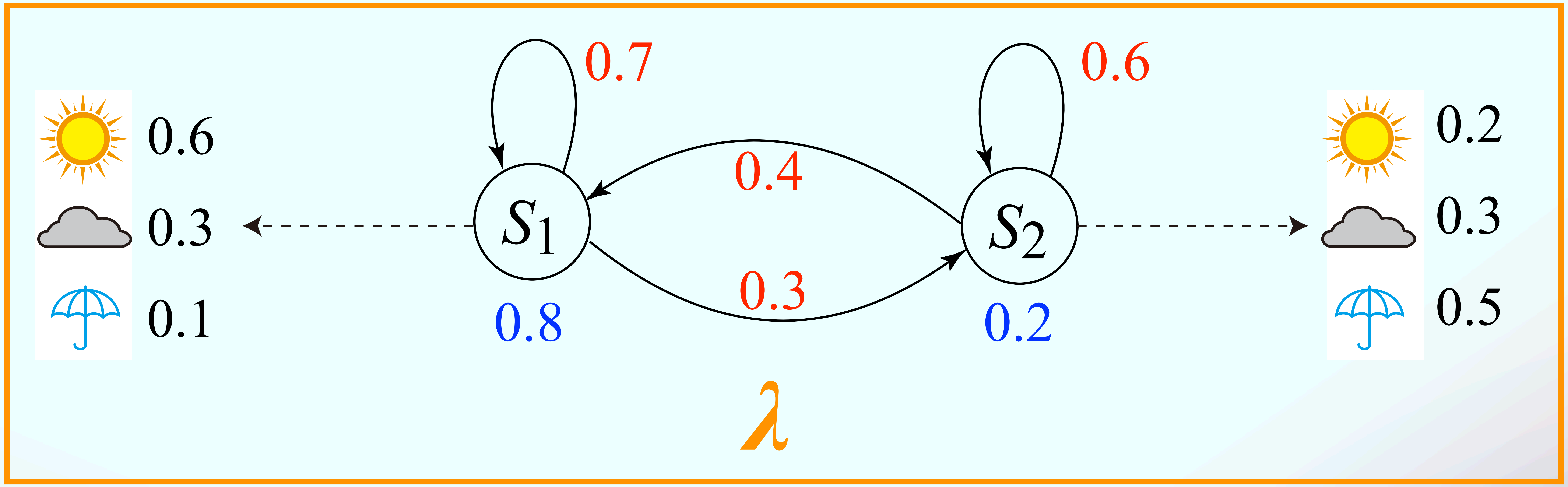
学習　観測系列 $\boldsymbol{O} = \boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T$ に対して

$P(\boldsymbol{O} \mid \lambda)$ を最大化するモデル λ を求める

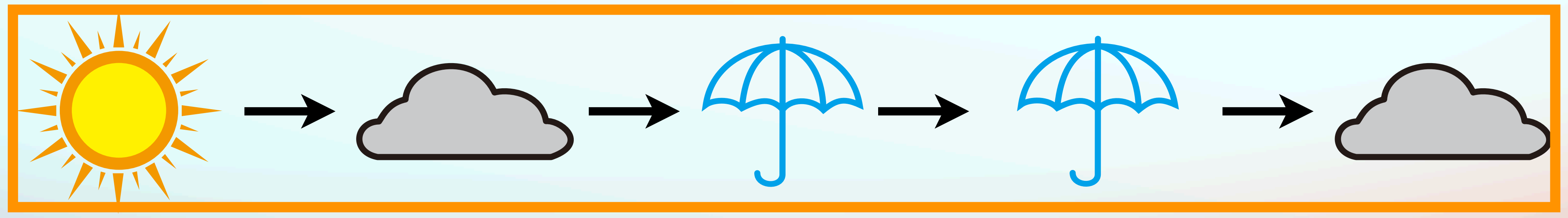
観測系列 $\boldsymbol{O}$ から状態系列 $\boldsymbol{Q}$ が一意に定まらないため、計算に工夫が必要。

状態系列 $\boldsymbol{Q} = q_1 q_2 q_3 \ldots q_T$

天気モデルの $\boldsymbol{O}$ と λ



$\boldsymbol{O}$

$P($  $| \lambda)=$

$0.8\times0.6\times0.7\times0.3\times0.3\times0.5\times0.6\times0.5\times0.4\times0.3$

$S_1 \quad S_1 \quad S_2 \quad S_2 \quad S_1$

$+ \quad 0.2\times0.2\times0.4\times0.3\times0.3\times0.5\times0.4\times0.1\times0.3\times0.3$

$S_2 \quad S_1 \quad S_2 \quad S_1 \quad S_2$

$=\dots$

認識　観測系列 $\boldsymbol{O}=\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T$ とモデル λ が与えられたとき

$P(\boldsymbol{O}|\lambda)$ を効率的に計算する

長さ T の全ての状態系列 $\boldsymbol{Q}=q_1, q_2, \ldots, q_T$ を枚挙し，

各状態系列に関する確率を累積してみると...

状態 q_1 の初期状態確率　状態 q_1 で観測記号 o_1 を出力する確率

$$P(O|\lambda) = \sum_{\boxed{\text{all } Q=q_1 q_2 \cdots q_T}} \boxed{\pi_{q_1}} \boxed{b_{q_1}(o_1)} \boxed{a_{q_1 q_2}} b_{q_2}(o_2) \cdots a_{q_{T-1} q_T} b_{q_T}(o_T) \qquad (5.3)$$

状態 q_1 から状態 q_2 へ遷移する確率

長さ T の全ての状態系列 $\boldsymbol{Q}$

状態系列 $\boldsymbol{Q}=q_1, q_2, \ldots, q_T$ の総数= N^T

各 $\boldsymbol{Q}$ について乗算 $(2T-1)N^T$ 回、

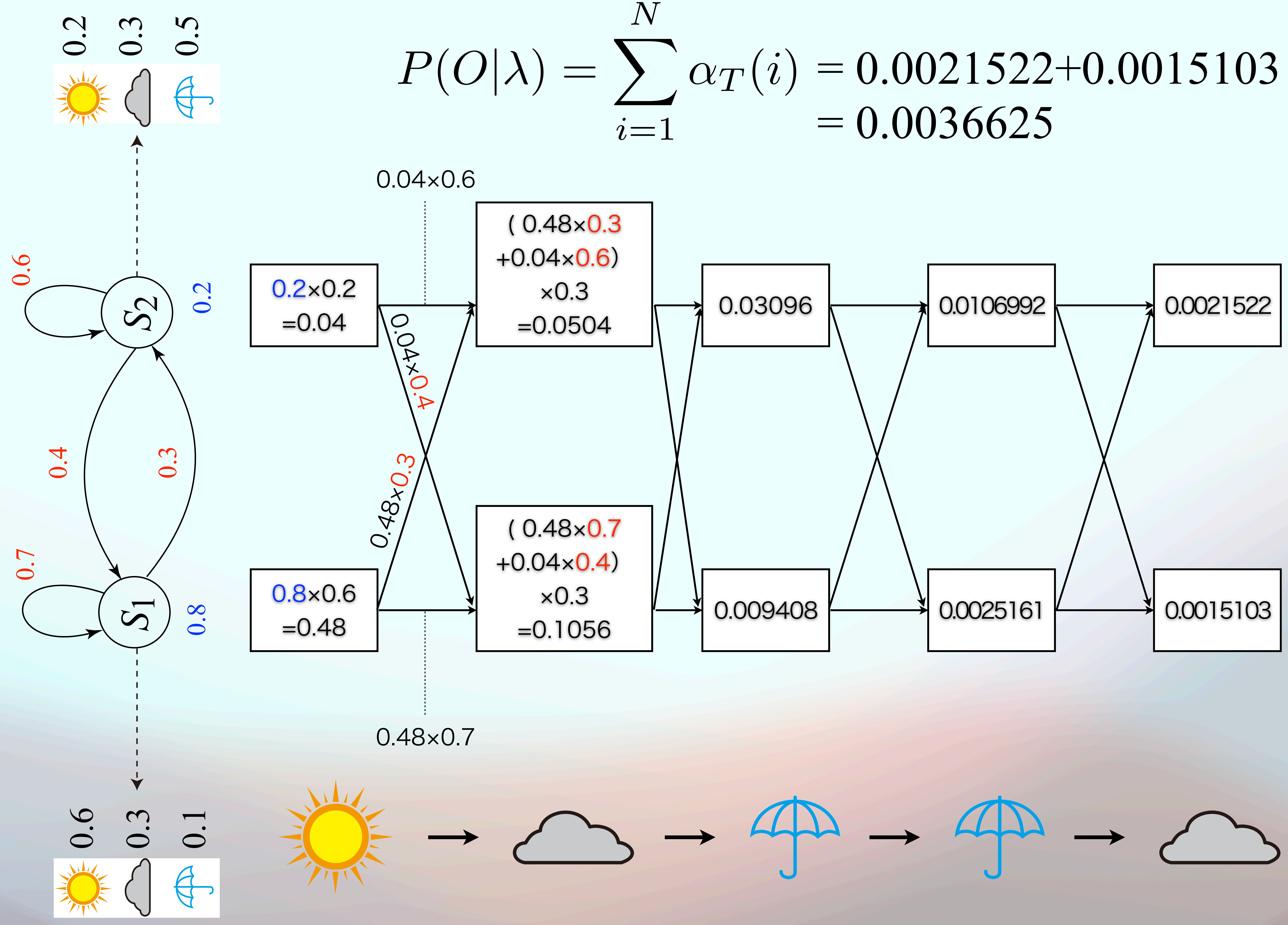
加算 N^T-1 回。

$N=3, \ T=50$

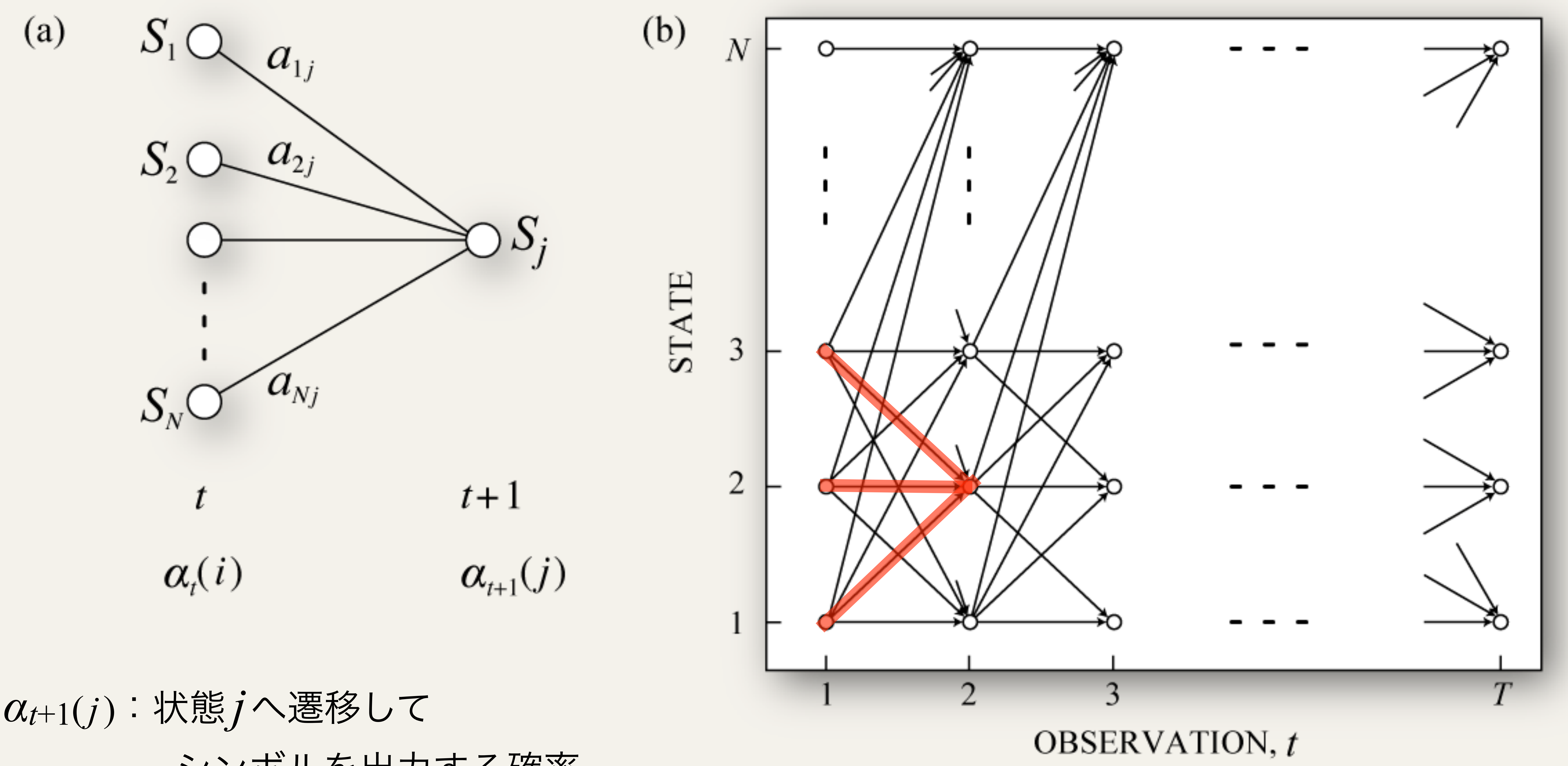
7.2×10^{23}

この例は3状態のHMMで0.5秒の音声の確率を計算することに相当...非現実的な計算量

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) = 0.0021522+0.0015103$$
$$= 0.0036625$$



動的計画法（部分問題の計算結果を記録しながら解いていく手法）で計算。



$\alpha_{t+1}(j)$：状態jへ遷移して
シンボルを出力する確率

trellis

N=3, T=50

$7.2\times10^{23} \rightarrow 885$

トレリス計算という。

動的計画法で計算量は10^{21}分の１に

前向きパスアルゴリズム　Forward Algorithm

1) 初期化

$$\alpha_1(i) = \pi_i b_i(\boldsymbol{o}_1), \quad 1 \le i \le N. \tag{5.5}$$

時刻 1 の状態 i の前向き確率 $\alpha_1(i)$ は状態 i の初期状態確率とシンボル $\boldsymbol{o}_1$ の出力確率の積

2) 帰納

$$\alpha_{t+1}(j) = \boxed{\left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right]} \boxed{b_j(\boldsymbol{o}_{t+1})}, \quad 1 \le t \le T-1, \quad 1 \le j \le N. \tag{5.6}$$

時刻 t+1 の状態 j でシンボル $\boldsymbol{o}_{t+1}$ を出力する確率

時刻 t の状態 i から時刻 t+1 の状態 j への遷移に伴う前向き確率 $\alpha_t(i)$ の積和

3) 停止

$$P(\boldsymbol{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{5.7}$$

最終時刻 T の前向き確率 $\alpha_T(i)$ の和が $P(\boldsymbol{O}\,|\,\lambda)$ である。

α を前向き確率という。(1) 初期化の後、状態 j と時刻 t を進めながら式 (5.6) によって前向き確率 α の積和の累積を進め、最終時刻 T の α の和を計算。

後ろ向きパスアルゴリズム Backward Algorithm

1) 初期化

$$\beta_T(i) = 1, \quad 1 \le i \le N. \tag{5.8}$$

2) 帰納

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j), \quad t = T-1, T-2, \ldots, 1, \quad 1 \le i \le N. \tag{5.9}$$

3) 停止

$$P(\boldsymbol{O}|\lambda) = \sum_{i=1}^{N} \pi_i b_i(\boldsymbol{o}_1) \beta_1(i). \tag{5.10}$$

β を後ろ向き確率という。(1) 初期化の後、状態 j と時刻 t を進めながら式 (5.9) によって後ろ向き確率 β の積和の累積を進め、最終時刻 T の α の和を計算。

実習5.6.3 p.5-15〜18 Forward・Backwardアルゴリズム

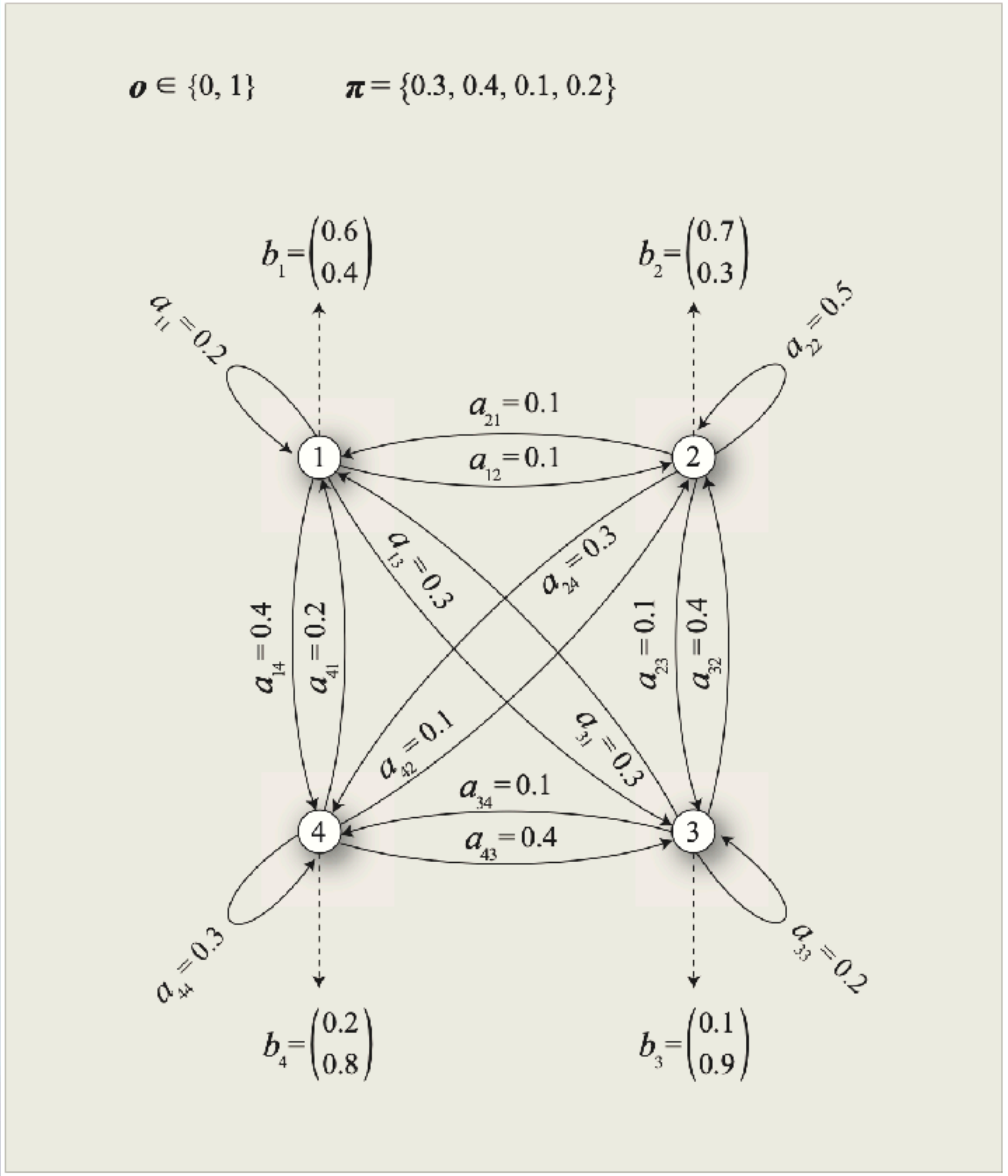


図 5.8: 4 状態離散出力 HMM

$$
\boldsymbol{A} = \begin{pmatrix} 0.2 & 0.1 & 0.3 & 0.4 \\ 0.1 & 0.5 & 0.1 & 0.3 \\ 0.3 & 0.4 & 0.2 & 0.1 \\ 0.2 & 0.1 & 0.4 & 0.3 \end{pmatrix}, \tag{5.44}
$$

$$
\boldsymbol{B} = \begin{pmatrix} 0.6 & 0.4 \\ 0.7 & 0.3 \\ 0.1 & 0.9 \\ 0.2 & 0.8 \end{pmatrix}, \tag{5.45}
$$

$$
\boldsymbol{\pi} = \{0.3, 0.4, 0.1, 0.2\}
$$

観測系列 $\boldsymbol{O} = 0\ 1\ 1\ 0$ に対する確率を

・Forwardアルゴリズム　・Backwardアルゴリズム

を用いて計算（各状態・時刻の確率も表示）

穴埋め課題

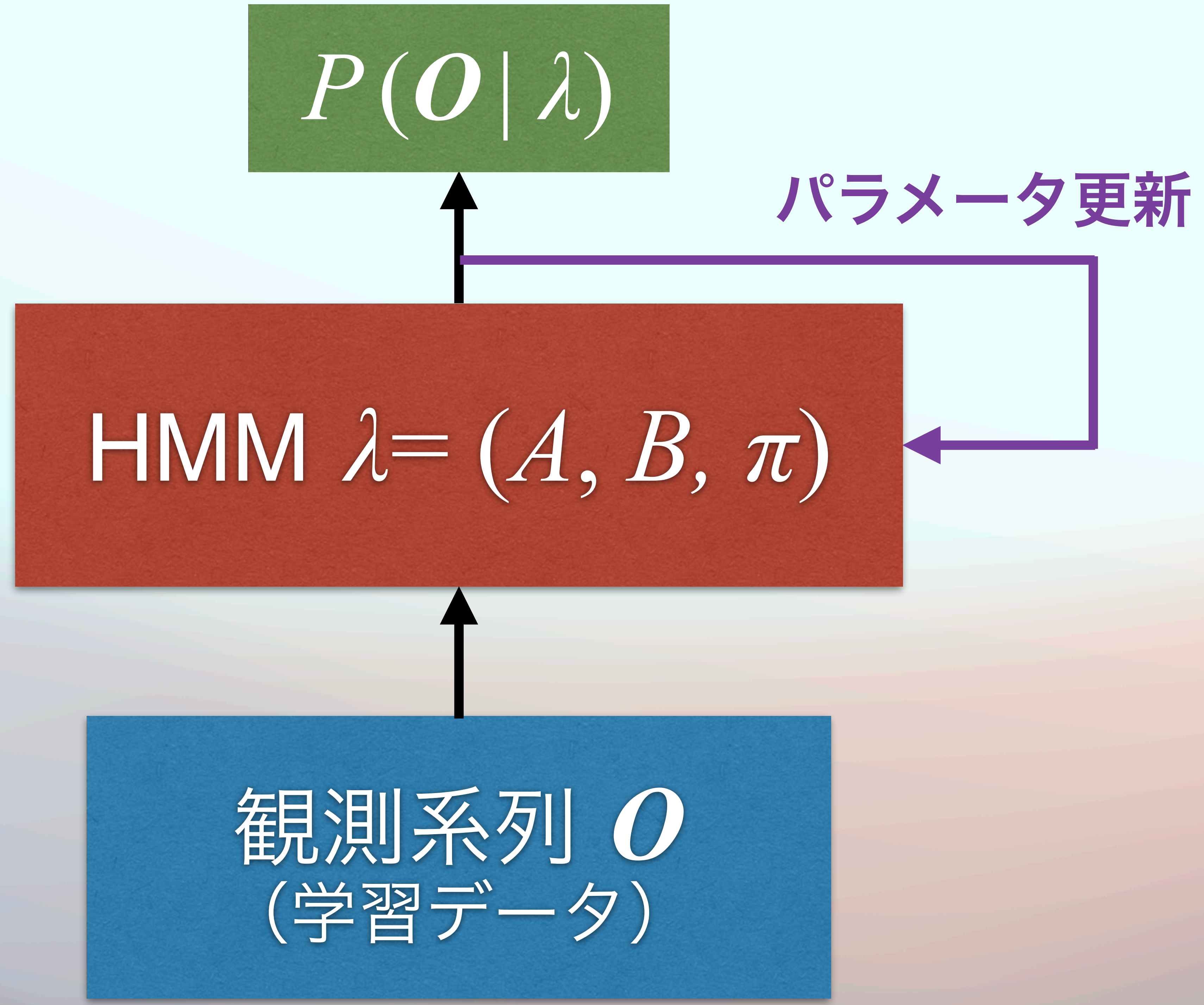
- **~/asr/drill/forward.c**
- **~/asr/drill/backward.c**

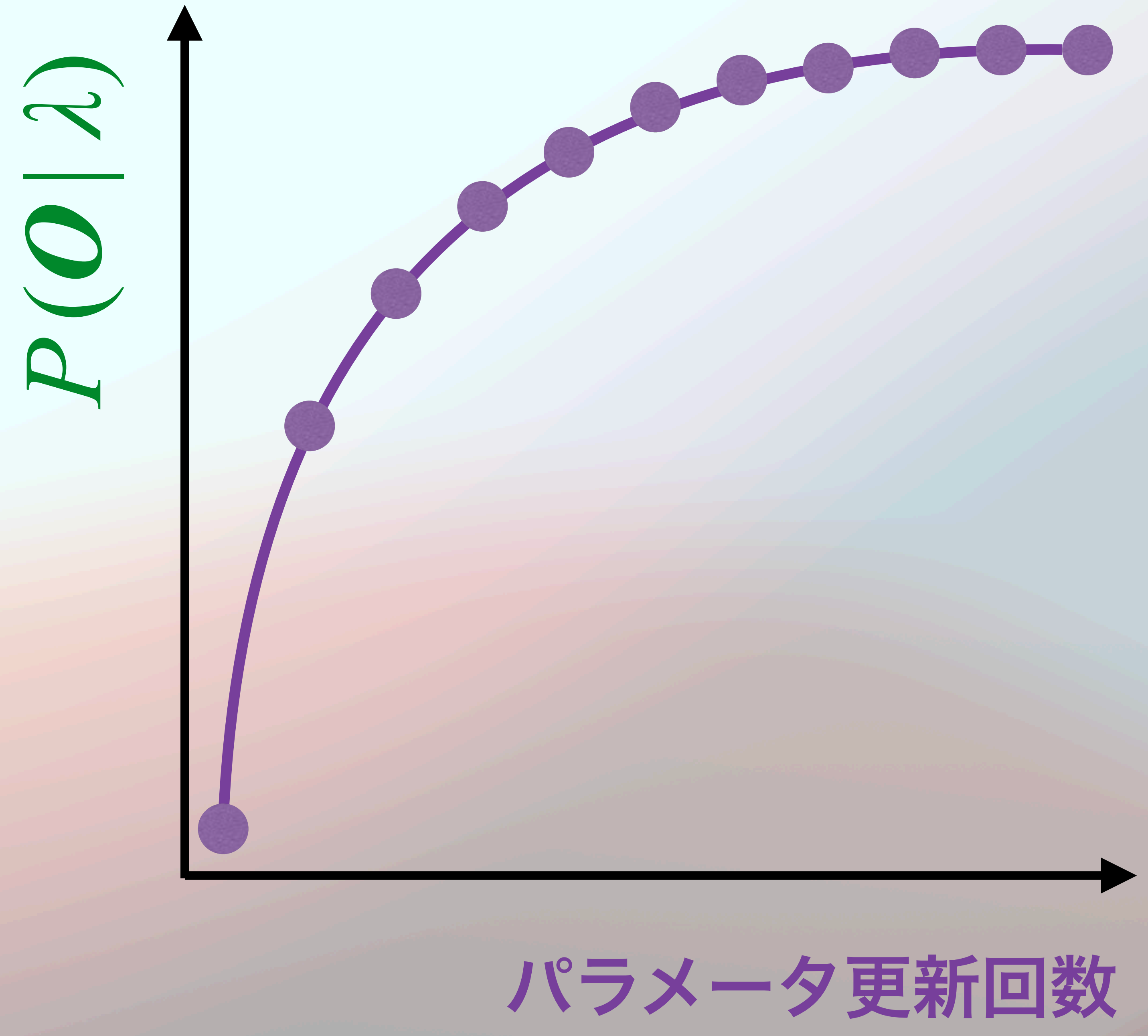
学習

観測系列 $\boldsymbol{O} = \boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T$ に対して

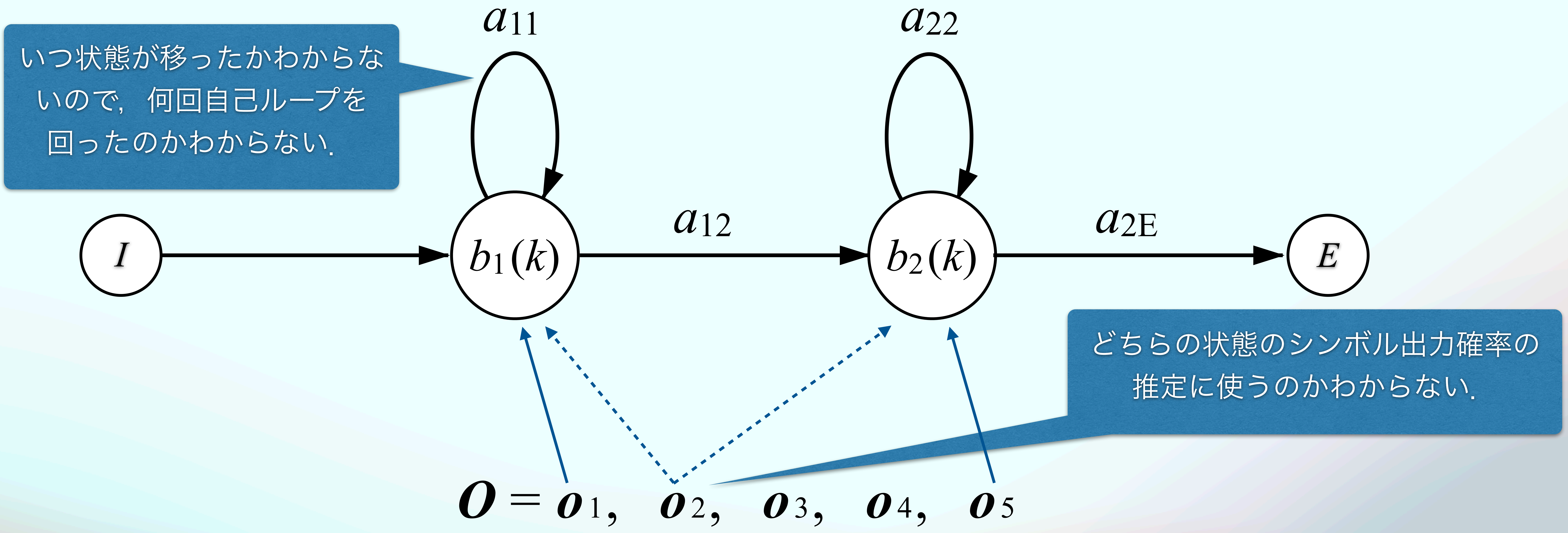
$P(\boldsymbol{O}\,|\,\lambda)$ を最大化するモデル λ を求める

観測データから母集団の確率分布を学習するにあたって、観測データとして直接観測できない潜在変数（隠れ変数）がある場合の学習方法の重要なものとして EM (expectation–maximization algorithm) アルゴリズムがある。





パラメータの学習　状態遷移確率 a_{ij} とシンボル出力確率 $b_i(k)$



問題点：学習データに対して状態遷移がわからない

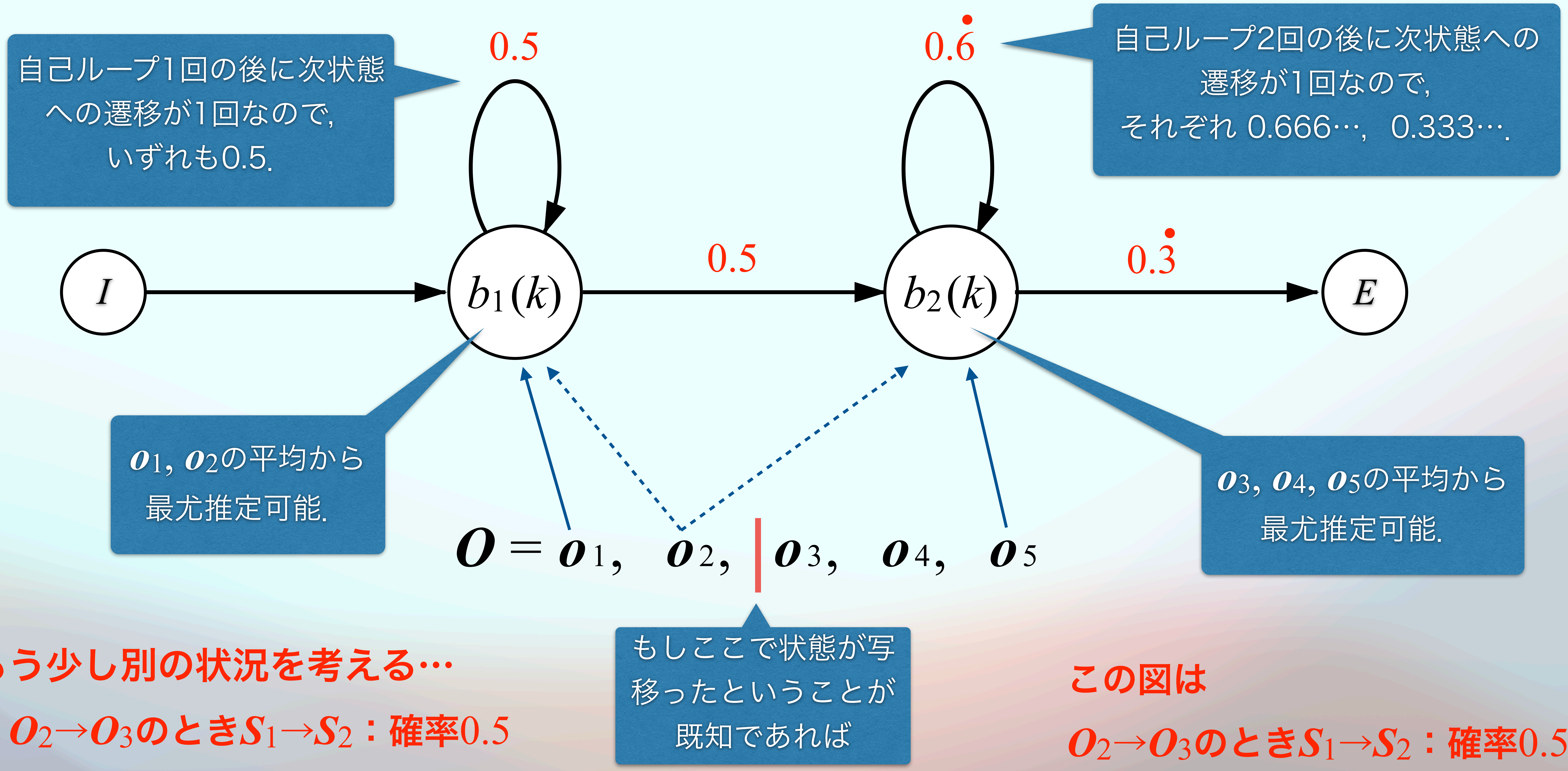
もし，状態遷移が既知であれば，

状態遷移確率は状態からの遷移の数え上げ，

シンボル出力確率は状態ごとに対応するシンボル出力平均から最尤推定することができる．

荒木雅弘，イラストで学ぶ音声認識，講談社，東京，2015年の解説による．

パラメータの学習　状態遷移確率 a_{ij} とシンボル出力確率 $b_i(k)$



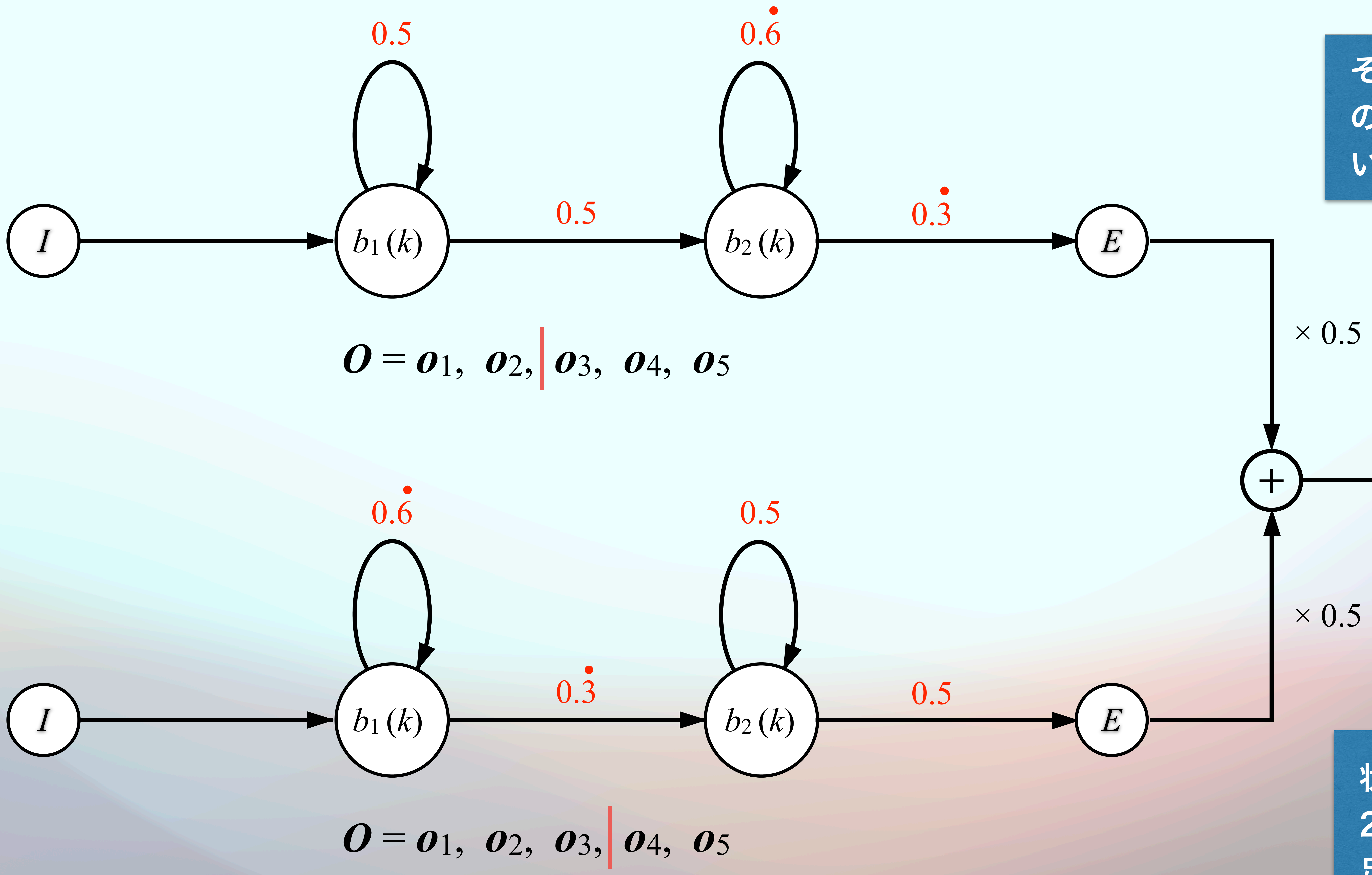
もう少し別の状況を考える…

- **$O_2 \to O_3$のとき$S_1 \to S_2$：確率0.5**
- **$O_3 \to O_4$のとき$S_1 \to S_2$：確率0.5**

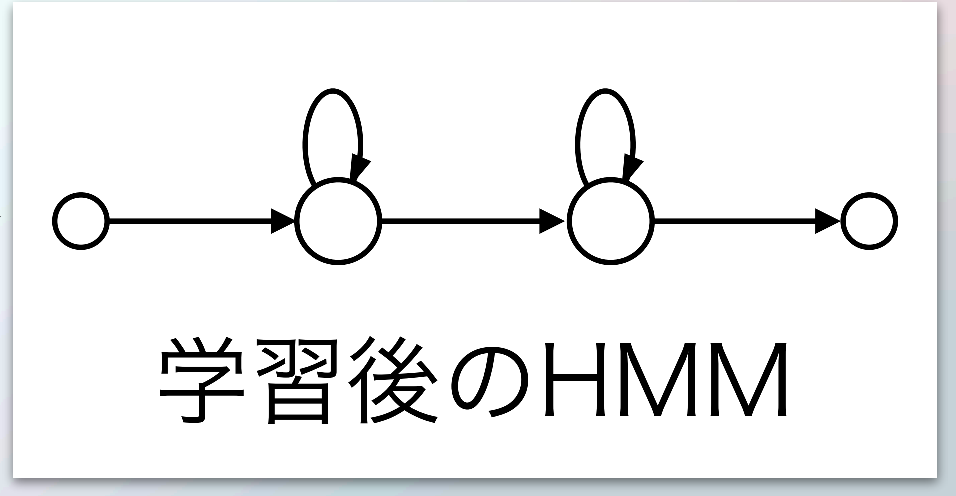
この図は
$O_2 \to O_3$のとき$S_1 \to S_2$：確率0.5の状況

荒木雅弘，イラストで学ぶ音声認識，講談社，東京，2015年の解説による.

パラメータの学習　状態遷移確率 a_{ij} とシンボル出力確率 $b_i(k)$



それぞれの状態系列の発生確率が0.5なので，学習パラメータの推定に寄与している割合も半々．



状態遷移系列の確率0.5を重みとして2つのHMMの対応するパラメータを足し合わせればよい

荒木雅弘，イラストで学ぶ音声認識，講談社，東京，2015年の解説による．

パラメータの学習　EMアルゴリズム Expectation–Maximization algorithm

1. HMMのパラメータを適当な初期値に設定.

2. Eステップ

(a) 学習データに対して，状態遷移を与えた時の確率を現在のHMMを用いて計算.

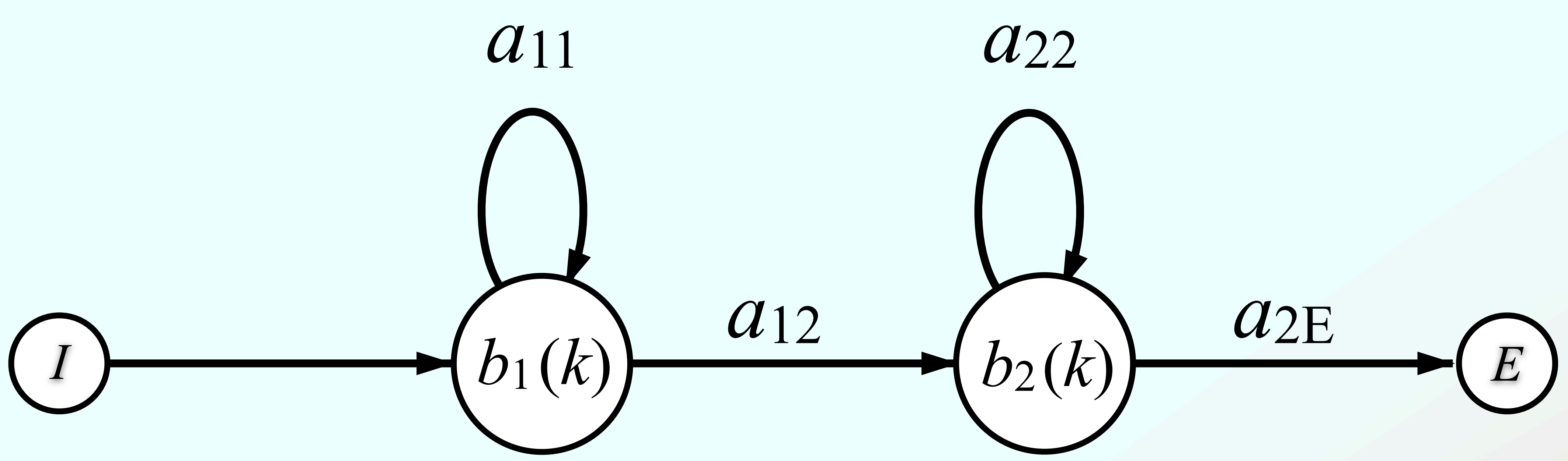
(b) それを全ての可能な状態遷移系列について求める.

3. Mステップ

(a) 全ての可能な状態遷移系列について，HMMのパラメータを最尤推定.

(b) Eステップで求めた状態遷移確率を重みとして最尤推定した結果を足し合わせる.

4. EステップとMステップを，Mステップのパラメータの変化量が一定値以下になるまで繰り返す.



Mステップ　Eステップで得られた確率を基にHMMを学習

$$P(\boldsymbol{o}_1, \boldsymbol{o}_2, \boldsymbol{o}_3, \boldsymbol{o}_4, \boldsymbol{o}_5)$$

$$P(\boldsymbol{o}_1, \boldsymbol{o}_2, \boldsymbol{o}_3, \boldsymbol{o}_4, \boldsymbol{o}_5)$$

$$P(\boldsymbol{o}_1, \boldsymbol{o}_2, \boldsymbol{o}_3, \boldsymbol{o}_4, \boldsymbol{o}_5)$$

$$P(\boldsymbol{o}_1, \boldsymbol{o}_2, \boldsymbol{o}_3, \boldsymbol{o}_4, \boldsymbol{o}_5)$$

Eステップ　Mステップで得られたHMMでそれぞれの状態遷移系列の確率を計算

荒木雅弘，イラストで学ぶ音声認識，講談社，東京，2015年の解説による.

パラメータの学習

1. HMMのパラメータを適当な初期値に設定.

2. Eステップ

(a) 学習データに対して，状態遷移を与えた時の確率を現在のHMMを用いて計算.

(b) それを全ての可能な状態遷移系列について求める.

3. Mステップ

(a) 全ての可能な状態遷移系列について，HMMのパラメータを最尤推定.

(b) Eステップで求めた状態遷移確率を重みとして最尤推定した結果を足し合わせる.

4. EステップとMステップを，Mステップのパラメータの変化量が一定値以下になるまで繰り返す.

Baum-Welch algorithm ある状態から最終状態までの確率をForwardおよびBackward確率で計算

Algorithm 1 Baum-Welch Algorithm

Baum-Welch Algorithm

$\bar{\lambda} = (\bar{\pi}_1, \ldots, \bar{\pi}_N, \bar{a}_{1,1}, \ldots, \bar{a}_{N,N}, \bar{b}_{1,1}, \ldots, \bar{b}_{N,M})$: パラメータ初期値 [入力]

$\boldsymbol{v}$: 出力記号 [入力]，　$o(1), o(2), \ldots, o(T)$: 記号列 [入力]

c: 収束判定用閾値 [入力]

$\bar{\lambda}$: パラメータ推定値 [出力]

$P(\lambda^{old}) \leftarrow -\infty$: 負の十分大きな値

START: $\lambda \leftarrow \bar{\lambda}$: パラメータの更新

Forward アルゴリズムにより λ の値から $\alpha_t(i)$ $(1 \le t \le T,\ 1 \le i \le N)$ と $P(\lambda)$ を計算する.

if $\log P(\lambda) - \log P(\lambda^{old}) < c$ **then**

　$\bar{\lambda} \leftarrow \lambda$: 推定結果

　exit

else

　$P(\lambda^{old}) \leftarrow P(\lambda)$

end if

Backward アルゴリズムにより λ の値から $\beta_t(i)$ $(1 \le t \le T,\ 1 \le i \le N)$ と $P(\lambda)$ を計算する.

for $i = 1$ to N **do**

$$\bar{\pi}_i = \frac{\alpha_1(i)\beta_1(i)}{P(\lambda)}$$

end for

for $i = 1$ to N **do**

　for $j = 1$ to N **do**

$$\bar{a}_{i,j} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{i,j} b_j(o(t+1)) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i)\beta_t(i)}$$

　end for

end for

for $j = 1$ to N **do**

　for $k = 1$ to M **do**

$$\bar{b}_{j,k} = \frac{\sum_{t=1}^{T} \delta_{v(k),o(t)} \alpha_t(j)\beta_t(j)}{\sum_{t=1}^{T} \alpha_t(j)\beta_t(j)}$$

$\delta_{v(k),o(t)}$ はクロネッカーのデルタ．$\delta_{x,y} = 1\,(x = y),\ 0\,(x \ne y)$.

　end for

end for

$\bar{\lambda} = (\bar{\pi}_1, \ldots, \bar{\pi}_N, \bar{a}_{1,1}, \ldots, \bar{a}_{N,N}, \bar{b}_{1,1}, \ldots, \bar{b}_{N,M})$

goto START

end of *Baum-Welch Algorithm*

Algorithm 1 Baum-Welch Algorithm

Baum-Welch Algorithm

$\bar{\lambda} = (\bar{\pi}_1, \ldots, \bar{\pi}_N, \bar{a}_{1,1}, \ldots, \bar{a}_{N,N}, \bar{b}_{1,1}, \ldots, \bar{b}_{N,M})$: パラメータ初期値 [入力]

$\boldsymbol{v}$: 出力記号 [入力], $o(1), o(2), \ldots, o(T)$: 記号列 [入力]

ϵ: 収束判定用閾値 [入力]

$\bar{\lambda}$: パラメータ推定値 [出力]

$P(\lambda^{old}) \leftarrow -\infty$: 負の十分大きな値

START: $\lambda \leftarrow \bar{\lambda}$: パラメータの更新

穴埋め部 (1)

Forward アルゴリズムにより λ の値から $\alpha_t(i)$ $(1 \leq t \leq T,\ 1 \leq i \leq N)$ と $P(\lambda)$ を計算する.

if $\log P(\lambda) - \log P(\lambda^{old}) < \epsilon$ **then**

    $\bar{\lambda} \leftarrow \lambda$: 推定結果

    exit

else

    $P(\lambda^{old}) \leftarrow P(\lambda)$

end if

穴埋め部 (2)

Backward アルゴリズムにより λ の値から $\beta_t(i)$ $(1 \leq t \leq T,\ 1 \leq i \leq N)$ と $P(\lambda)$ を計算する.

for $i = 1$ **to** N **do**

end if

Backward アルゴリズムにより λ の値から $\beta_t(i)$ $(1 \leq t \leq T,\ 1 \leq i \leq N)$ と $P(\lambda)$ を計算する．

for $i = 1$ to N **do**

$$\boxed{\bar{\pi}_i = \frac{\alpha_1(i)\beta_1(i)}{P(\lambda)}}$$

穴埋め部 (3)

end for

for $i = 1$ to N **do**

  for $j = 1$ to N **do**

$$\bar{a}_{i,j} = \boxed{\frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{i,j} b_j(o(t+1)) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i)\beta_t(i)}}$$

穴埋め部 (4)

  end for

end for

for $j = 1$ to N **do**

$$\sum_{t=1} \alpha_t(i)\beta_t(j)$$

```
    end for
end for
for j = 1 to N do
    for k = 1 to M do
```

$$\bar{b}_{j,k} = \boxed{\frac{\sum_{t=1}^{T} \delta_{v(k),o(t)}\alpha_t(j)\beta_t(j)}{\sum_{t=1}^{T} \alpha_t(j)\beta_t(j)}}$$

穴埋め部 (5)

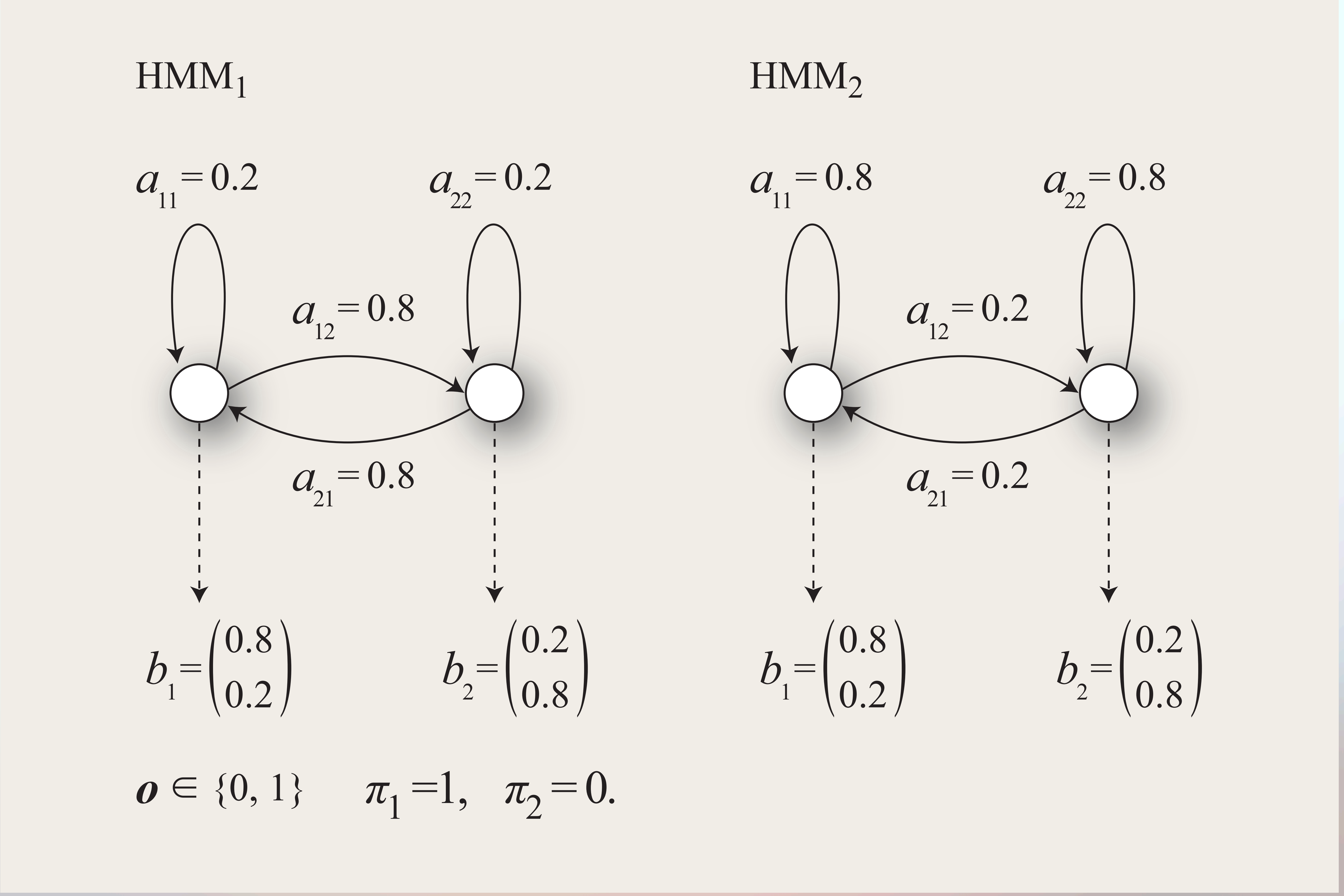
$\delta_{v(k),o(t)}$ はクロネッカーのデルタ． $\delta_{x,y} = 1\,(x = y),\ 0\,(x \neq y)$.

「$v(k)$=$o(t)$であれば足す。でなければ足さない。」の実装方法

1. $\delta_{x,y}$ を定義　(a)マクロ（**#define**）　(b) 関数
2. 条件判断 **if-then-else**
3. 三項演算 **? :**

実習5.6.3 p.5-18 — HMMの学習と認識

`~/asr/drill/baumwelch.c` の穴埋め課題



０ と １ の系列を出力／認識する隠れマルコフモデル

HMMは記号系列の**生成**器としても**認識**器としても使うことができる


HMM1
a11=0.2 a22=0.2 a12=0.8 a21=0.8
b1=(0.8 0.2) b2=(0.2 0.8)
001101011100.....
1000個
1001110011001.....
HMM2
a11=0.8 a22=0.8 a12=0.2 a21=0.2
b1=(0.8 0.2) b2=(0.2 0.8)
Initial HMM0
a11=0.5 a22=0.5 a12=0.5 a21=0.5
b1=(0.5 0.5) b2=(0.5 0.5)
Baum-Welchアルゴリズム
② 学習データから
信号源の統計的性質を
HMMとして学習
① 2種類の信号源から
学習データを生成
③ 学習済のHMMは
信号源のHMMに類似
H̃MM1
a11=0.23 a22=0.23 a12=0.81 a21=0.79
b1=(0.74 0.27) b2=(0.27 0.74)
H̃MM2
a11=0.77 a22=0.82 a12=0.23 a21=0.18
b1=(0.82 0.18) b2=(0.18 0.82)

パラメータの学習の様子

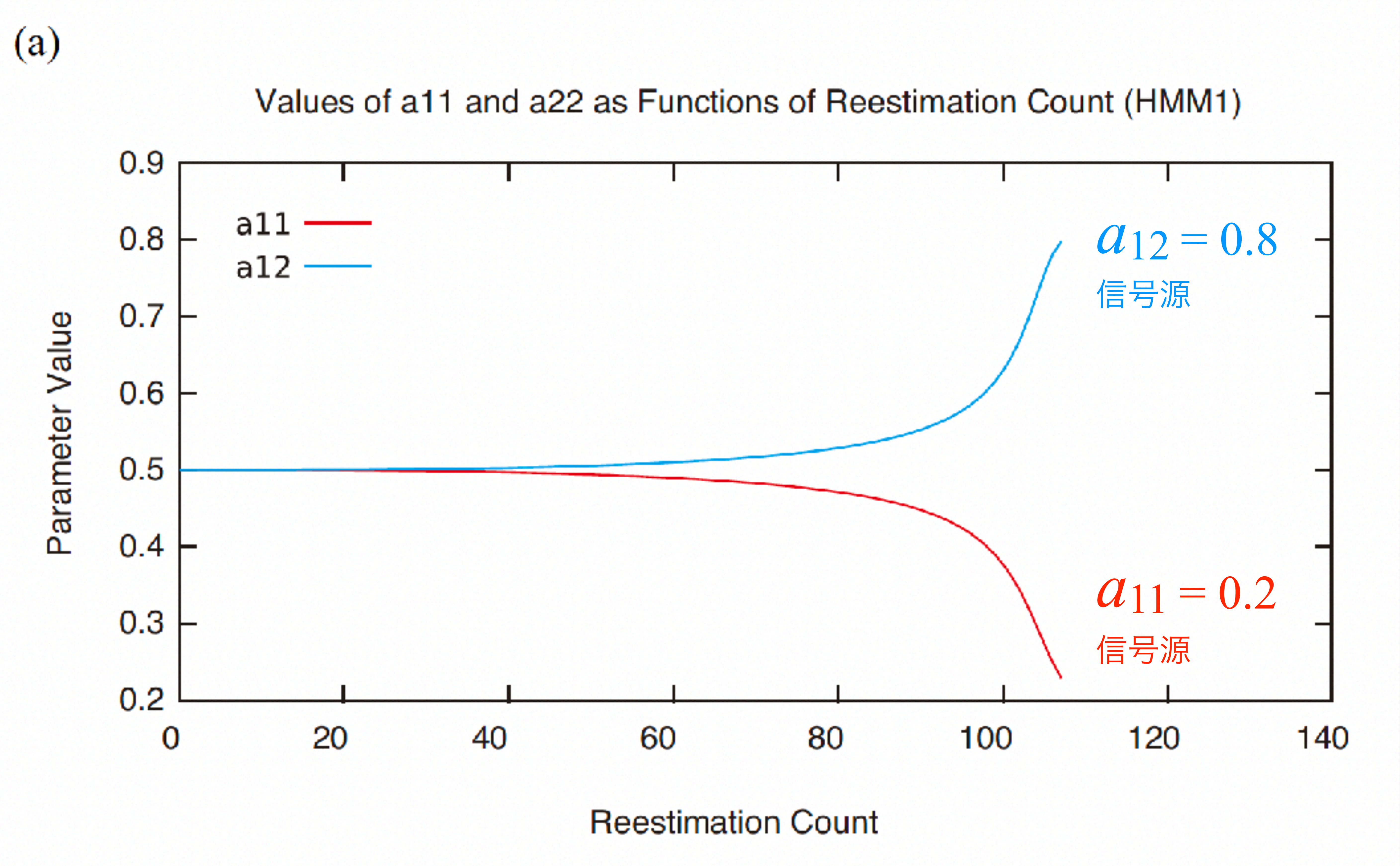
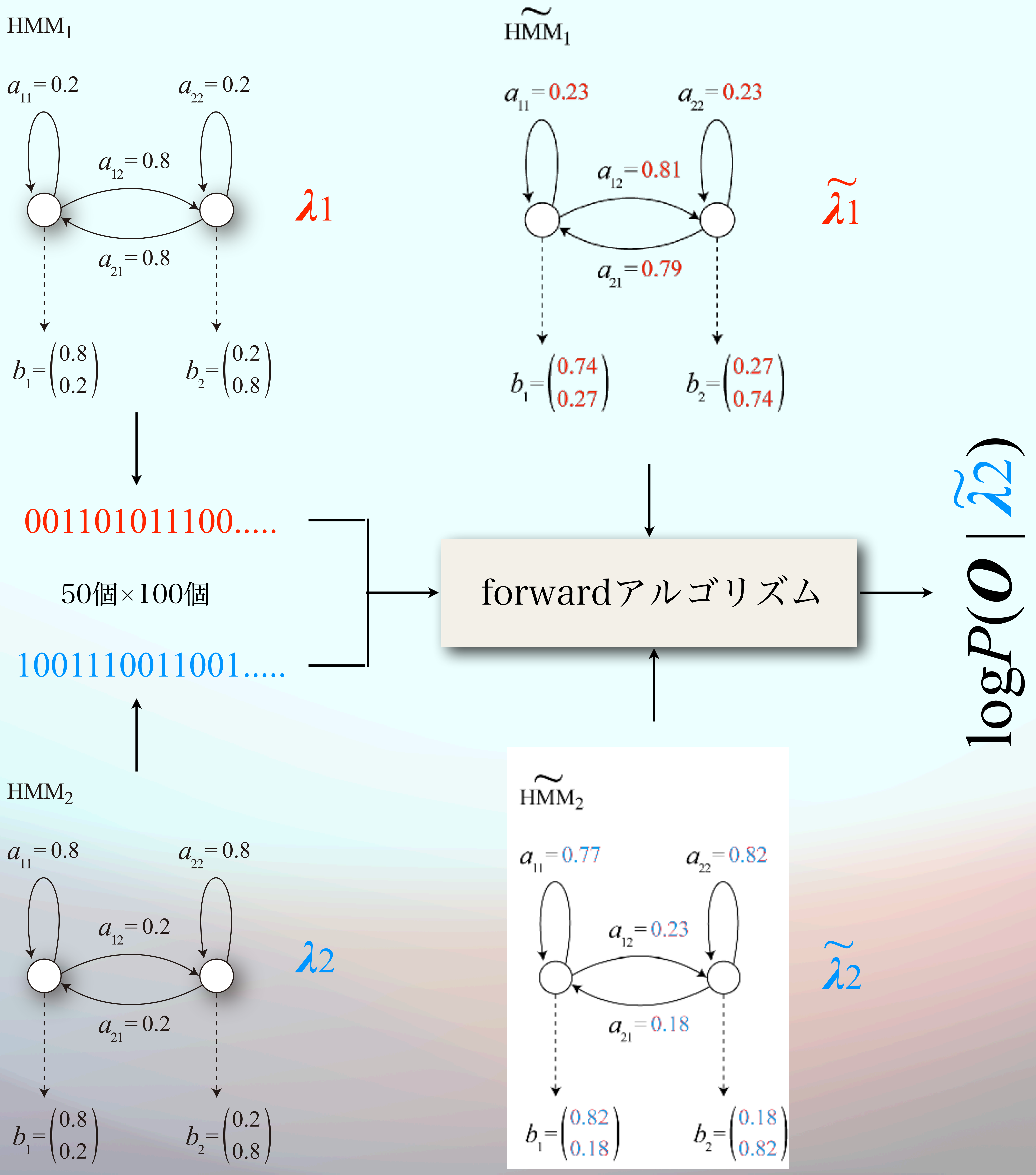
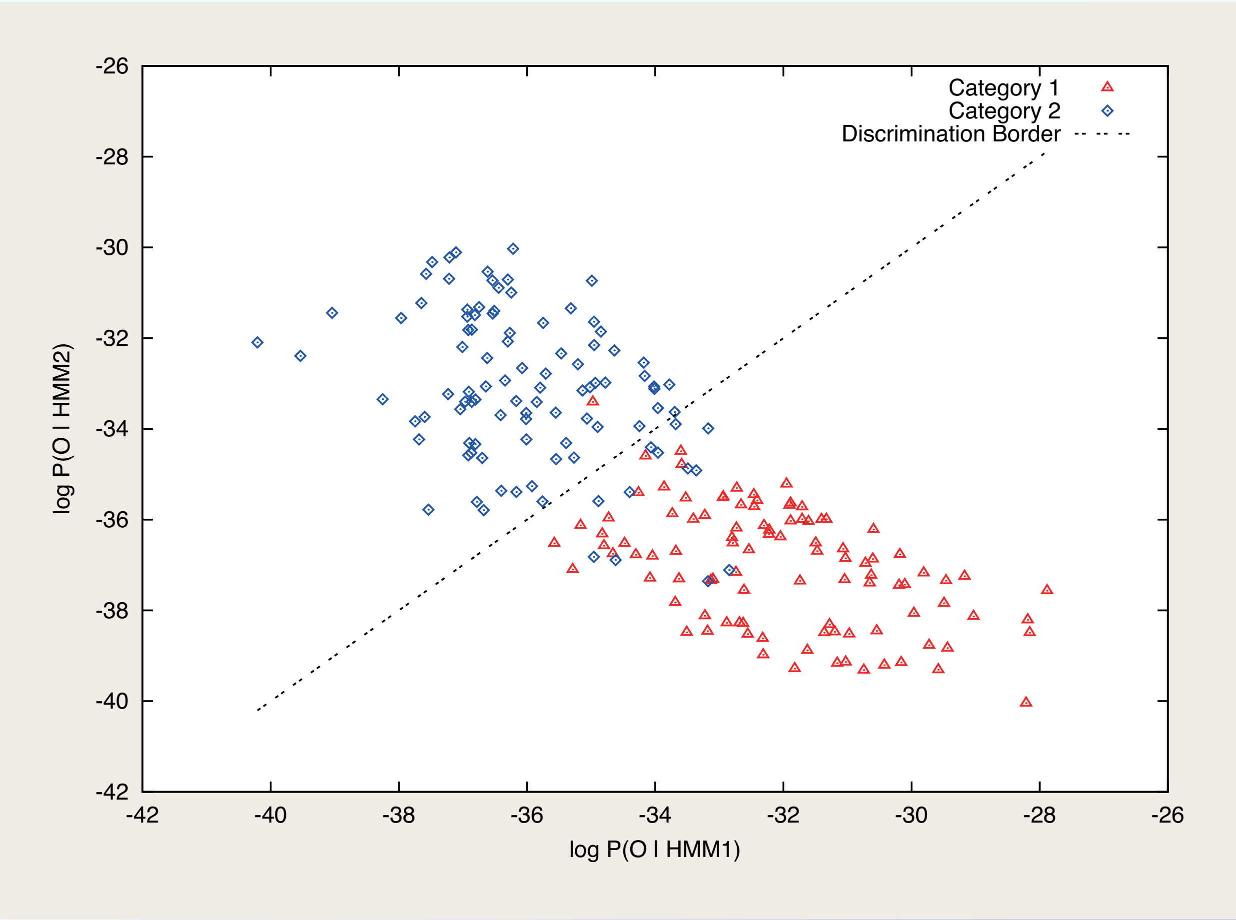


図 5.13: (a) HMM1の状態遷移確率 a_{11} と a_{12} が再推定の繰り返しにより信号源のパラメータに近づく．

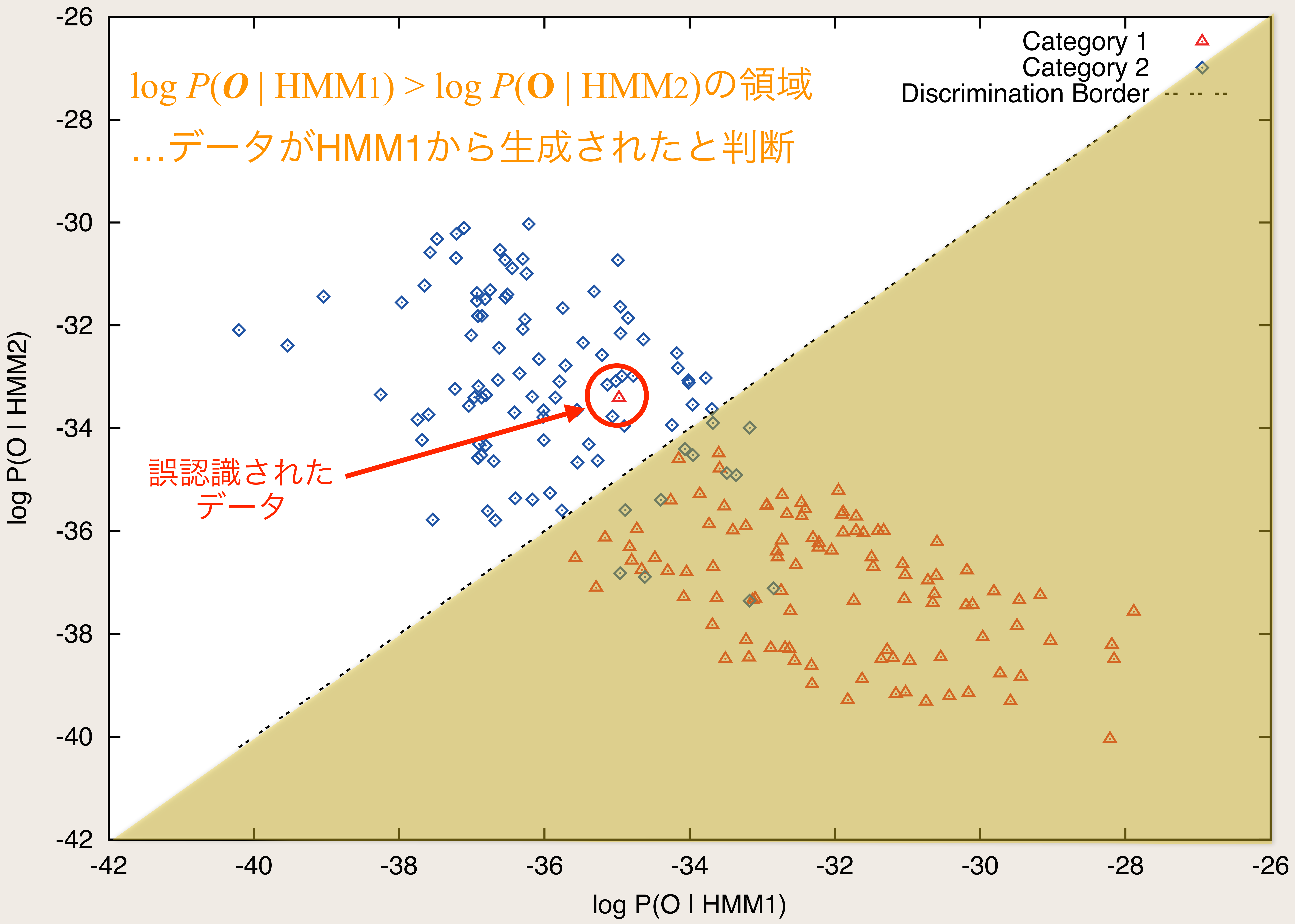
認識

学習したHMMを使って
信号がどちらの信号源からの
ものか判断する



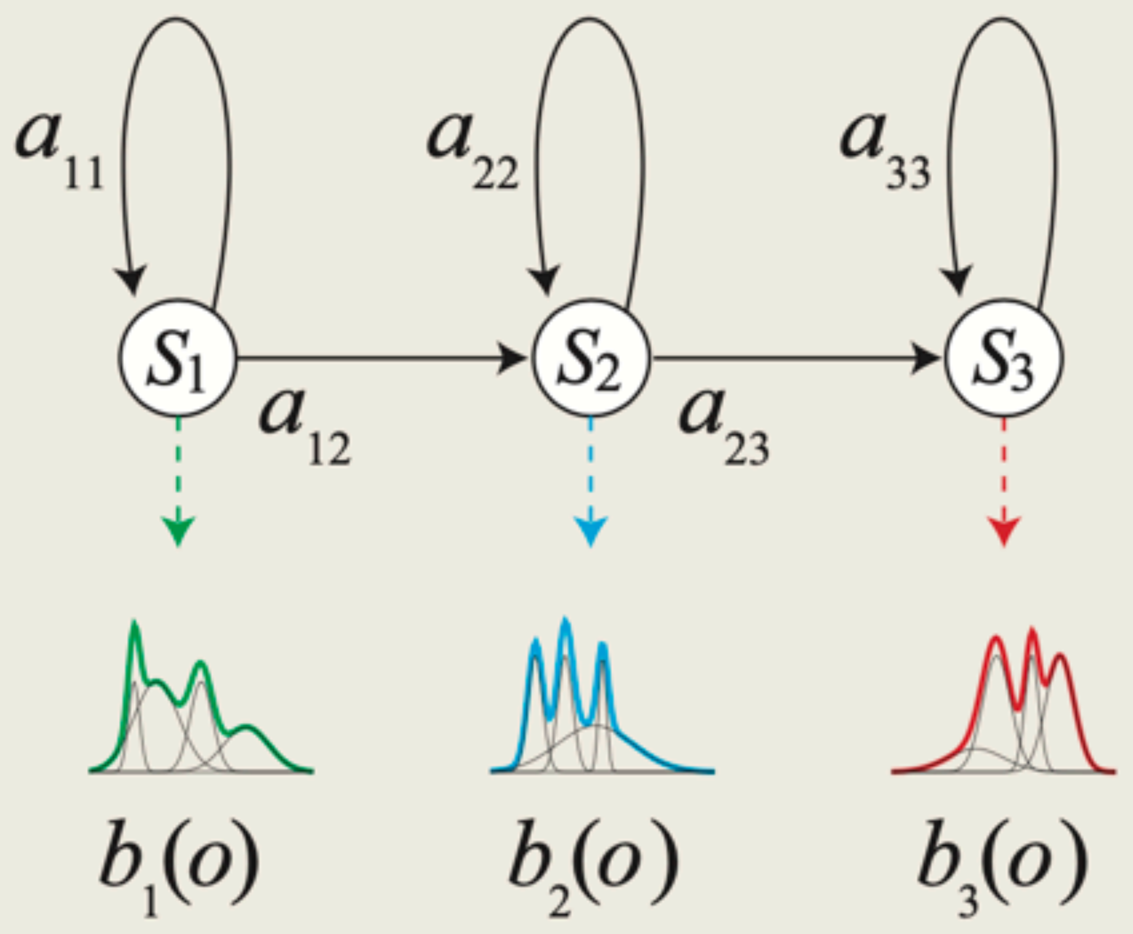


$\log P(\boldsymbol{O} \mid \tilde{\lambda}1)$



Category 1
Category 2
Discrimination Border
log P(O | HMM1) > log P(O | HMM2)の領域
...データがHMM1から生成されたと判断
誤認識された
データ
log P(O | HMM2)
log P(O | HMM1)

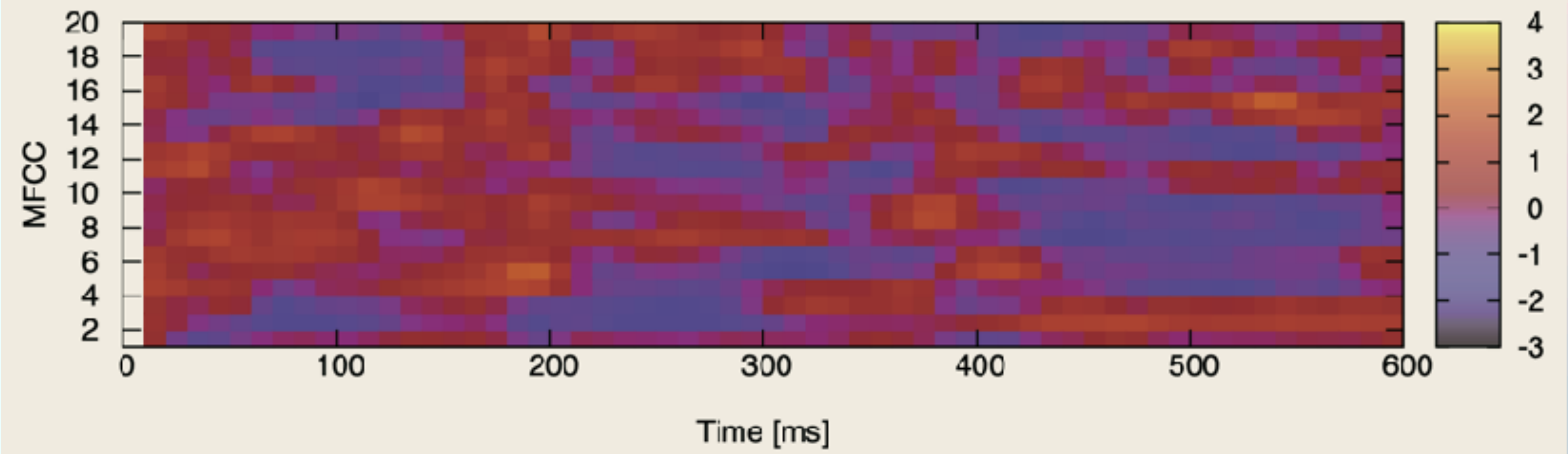
実習5.6.4 p.5-25 — 多次元ガウス確率密度関数



$$b_j(\boldsymbol{o}_t) = \sum_{m=1}^{M} c_{jm}\, \mathcal{N}[\boldsymbol{o}_t, \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}], \quad 1 \le j \le N.$$

図 5.7: 出力確率密度が混合ガウス分布である連続 HMM

- 単語HMMの学習と認識に必要
- **~/asr/wrecog/program/gpdf.c** の穴埋め課題…式(5.35)の実装



単語認識用音響モデル $P(\mathbf{O}|\lambda)$ としてHMMでモデル化する

観測系列 $\mathbf{O}$ はMFCCのような連続値（実数ベクトル）

$$\mathcal{N}(\boldsymbol{o}_t, \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}) = \frac{1}{\sqrt{\prod_{d=1}^{D} 2\pi\sigma_d^2}} \exp\left\{-\frac{1}{2}\sum_{d=1}^{D}\frac{(\boldsymbol{o}_t^{(d)} - \boldsymbol{\mu}_m^{(d)})^2}{\sigma_d^2}\right\} \qquad (5.35)$$

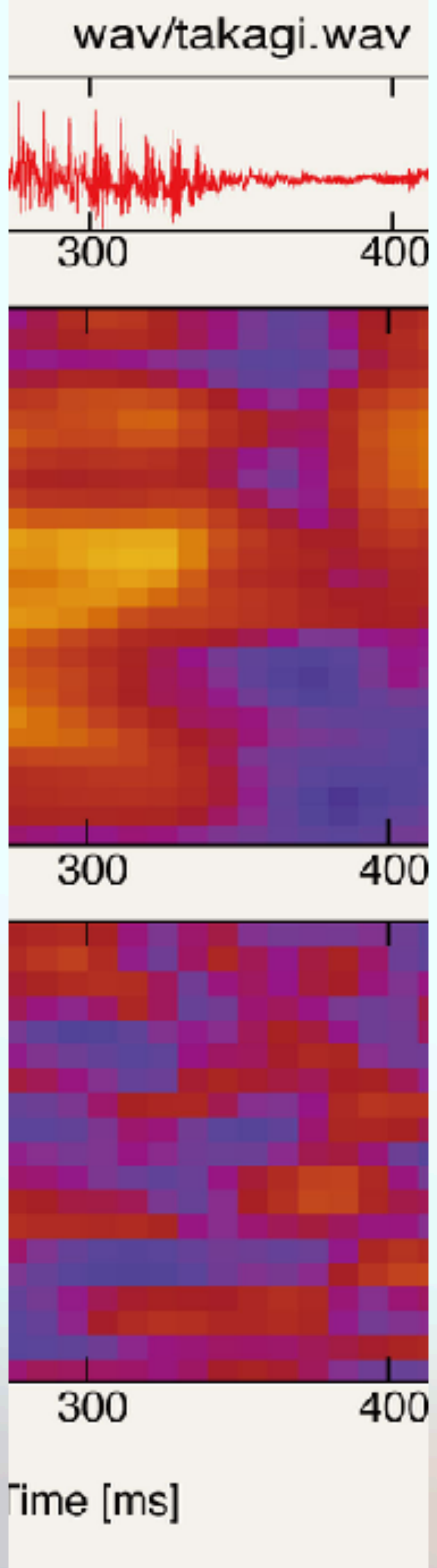
Q: なぜフィルタバンク出力を離散コサイン変換してMFCCにする？

A: MFCCにすると次元間の相関が無くなりスペクトル分布のモデル化が容易になる

フィルタバンク出力

離散コサイン変換

メル周波数ケプストラム係数
MFCC



特徴量なので，多次元ガウス分布の確率密度関数を使います．

$$\mathcal{N}(\boldsymbol{o}_t, \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}) = \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}_m|}} \exp\left\{-\frac{1}{2}(\boldsymbol{o}_t - \boldsymbol{\mu}_m)'\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{o}_t - \boldsymbol{\mu}_m)\right\} \quad (5.33)$$

であり，D は $\boldsymbol{o}_t$ の次数，$'$ は転置，M は混合数，$\boldsymbol{\Sigma}_m^{-1}$ は $\boldsymbol{\Sigma}_m$ の逆行列，$|\boldsymbol{\Sigma}_m|$ は $\boldsymbol{\Sigma}_m$ の行列式を表わします．多次元データを利用した場合，$|\boldsymbol{\Sigma}_m|$ は共分散行列となりますが，この実験では，次元間の相関が無いと仮定した対角共分散行列を使用します．この場合，共分散行列は

相関が有ると仮定すると非対角成分を推定しなければならず、膨大な学習データが必要となる。実用上は対角共分散で十分。

$$\boldsymbol{\Sigma}_m = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ 0 & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & \sigma_D^2 \end{pmatrix} \quad (5.34)$$

となり，密度関数は，

$$\mathcal{N}(\boldsymbol{o}_t, \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}) = \frac{1}{\sqrt{\prod_{d=1}^{D} 2\pi\sigma_d^2}} \exp\left\{-\frac{1}{2}\sum_{d=1}^{D}\frac{(o_t^{(d)} - \mu_m^{(d)})^2}{\sigma_d^2}\right\} \quad (5.35)$$

で与えられます．ここで，$\boldsymbol{o}_t^{(d)}$ は観測ベクトル $\boldsymbol{o}_t$ の第 d 次元，$\boldsymbol{\mu}_m^{(d)}$ は第 m 混合の平均値ベクトル $\boldsymbol{\mu}_m$ の第 d 次元です．

本日の実習

この実習の実験はtcshで実行
コマンドラインで「tcsh」と入力するとシェルがtcshに切り替わる

本日の実習手順

- §5.6.1 drillF ［実習］forward アルゴリズム
- §5.6.2 drillB ［実習］backward アルゴリズム
- §5.6.3 ［実習］Baum-Welch アルゴリズム
- p.5-21 drillT ［実習］HMM の学習実験
- p.5-23 drillR ［実習］HMM による認識実験
- §5.6.4 drillG ［実習］多次元ガウス確率密度関数

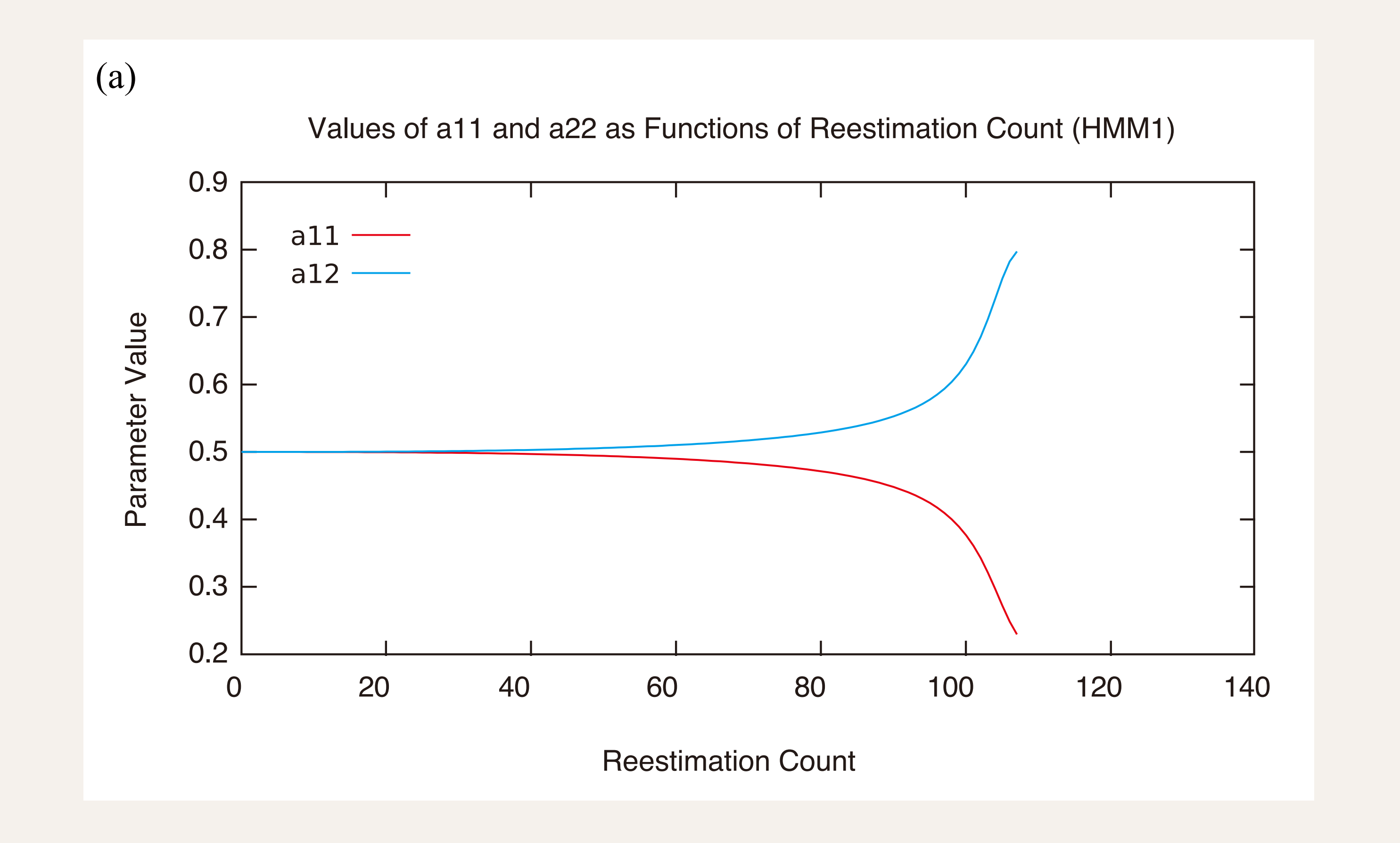


図 5.13(a): HMM_1 の状態遷移確率 a_{11} と a_{12} が再推定の繰り返しにより信号源のパラメータに近づく様子．

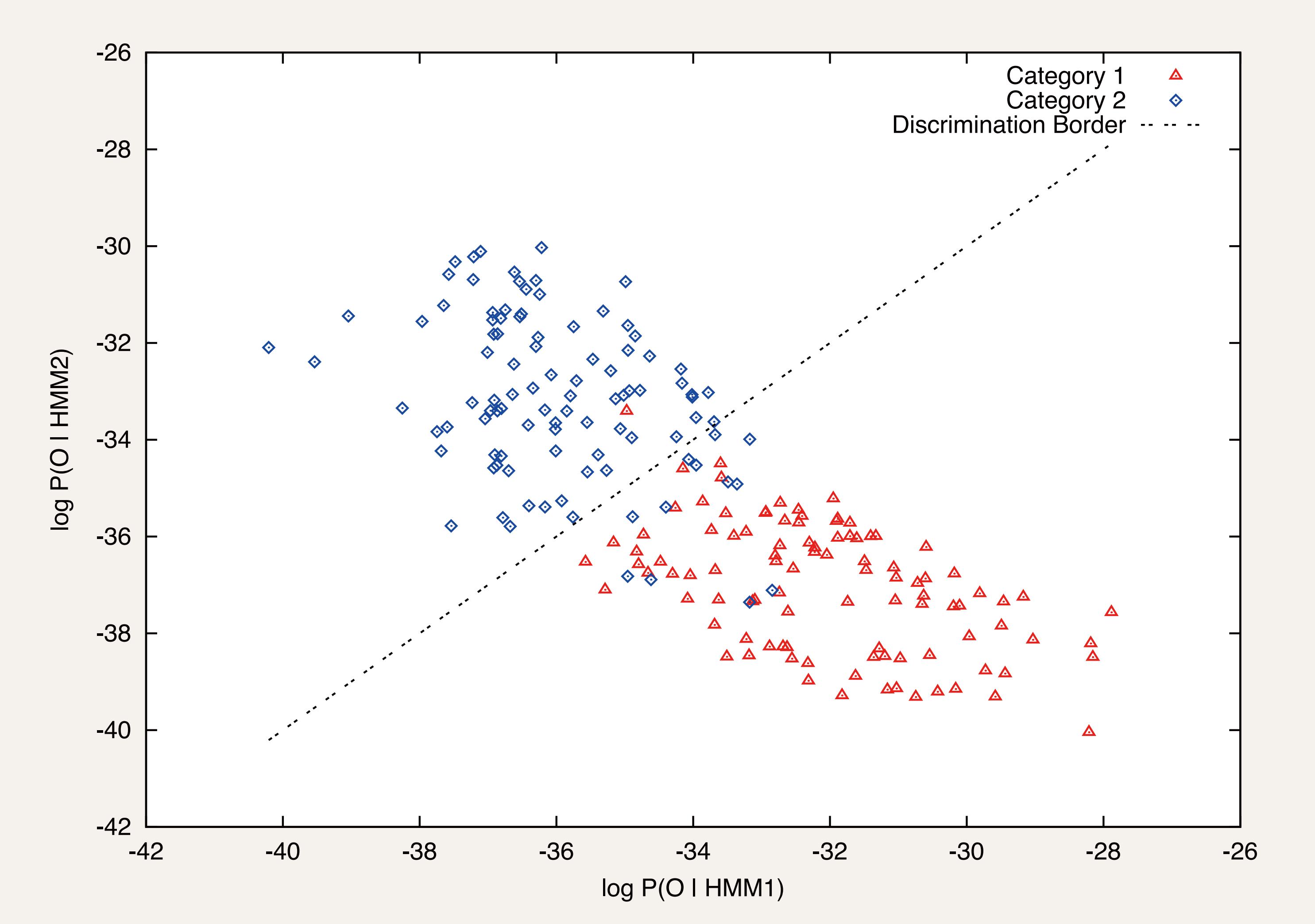


図 5.12　２カテゴリのパターン認識．点線は識別平面．

本日も時間が余ったら…

- 単語音声データ（3日目）の録音
  - 認識対象単語が決まっていれば
  - 手順はテキスト第6章参照

【理由】

- 単語音声の学習データ作成では、ラベル付け作業がかなりの手間（約90分）。
- 録音だけは早めに行っておく。
- 余裕があればラベル付けを始める。