東京大学 工学部 計数工学科/物理工学科

応用音響学：Viterbi アルゴリズム

嵯峨山 茂樹 ＜sagayama@hil.t.u-tokyo.ac.jp＞
東京大学 工学部 計数工学科 http://hil.t.u-tokyo.ac.jp/

■ 参考文献

- ■ 鹿野, 中村, 伊勢, 「音声・音情報のデジタル信号処理」, 昭晃堂, 1997.
- ■ 中川 聖一「確率モデルによる音声認識」コロナ社
- ■ 古井 貞熙「音声情報処理」森北出版
- ■ 谷萩 隆嗣「音声と画像のディジタル信号処理」コロナ社

音声認識の手法

- 音声認識
  - 非線形パターンマッチング
  - 確率モデル (特に 隠れマルコフモデル) → 現代の主流
  - 神経回路網 (ニューラルネットワーク)
  - 人工知能的アプローチ (エキスパート，知識ベース)
- 不特定話者音声認識と話者適応
- 構文解析アルゴリズムと言語モデル

連続音声認識の構成/原理

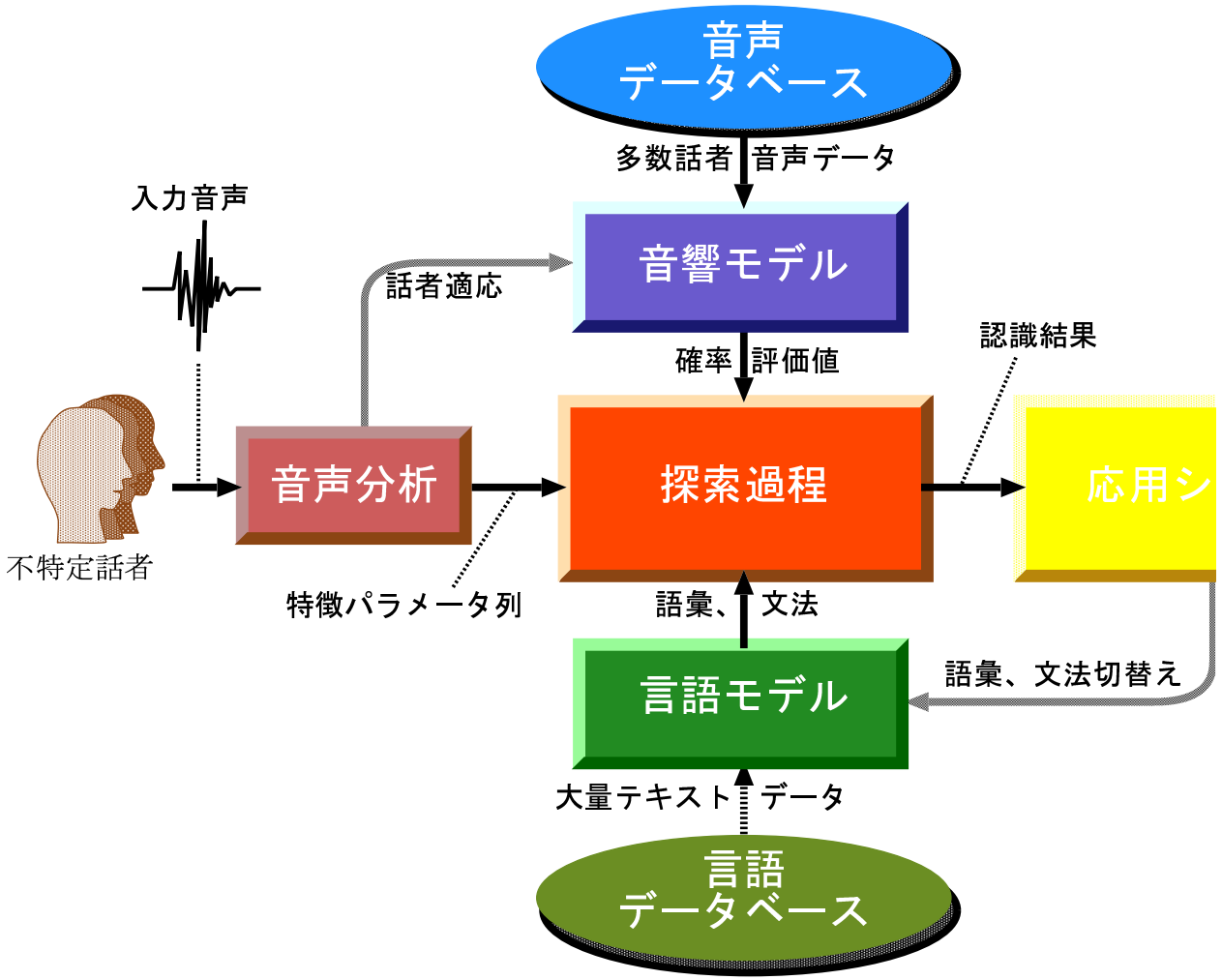


図1. 連続音声認識の構成要素。音声分析により音声特徴を抽出し、語彙や文法によって音素の並びを規定する言語モデルのもとで、音素を確率モデルにより表現する音響モデルによって入力の確率評価をして、事後確率最大の経路を効率良く探索する。

確率モデル(HMM)を用いた音声認識手法

- 「隠れマルコフモデル」(hidden Markov model: HMM)
- 確率モデル → 音声パターンの統計的変動がうまく記述できる
- 確率モデル → 個々の事象の確率の積/和で音声全体の確率が定義できる
- 3種の基本アルゴリズム(計算効率が高い)が揃っている

1. 観測確率問題 — Forward algorithm
   モデルから観測音声が生成される確率が得られる.
2. 最適経路問題 — Viterbi decoding algorithm
   観測音声を生成する確率最大のモデル内部経路が得られる.
3. 最適学習問題 — Baum-Welch reestimation algorithm
   複数の観測音声を生成する確率が最大となるモデルパラメータが求められる.

Hidden Morkov Model (HMM)

HMMの特徴

- 一般的に大きな変動のある特徴ベクトルの時系列を確率モデルで表現。
- 各単語や音素を標準的な時系列を、DPのように標準パターンとして用いず、確率状態遷移機械(マルコフモデル)で表現する。
- 事後確率 $P(O|\lambda)$ を求めることを目的とした方法。

HMMの利点

- 特徴ベクトルの時系列パターンの統計的変動を最も吸収するように、モデルのパラメータを推定できる。
- 認識における計算量が少ない。

HMMの問題点

- モデルの設計法が確立されていない。
- 統計的モデルであるので、モデルのパラメータ推定にある程度のサンプルを必要とし、計算量も多い。

HMMの3つの基本問題

モデル$\lambda = (\overset{\text{状態遷移確率行列}}{A}, \overset{\text{観測シンボル確率分布}}{B}, \overset{\text{初期状態分布}}{\pi})$と観測系列$Y = (y_1, y_2, \ldots, y_T)$が与えられたとき

問題1 (確率評価)
モデルλに対する観測系列Yの確率$P(Y|\lambda)$の計算

- モデルλが観測系列Yに対してどの程度適応しているか

$\Rightarrow$ 例：フォワードパスアルゴリズム

問題2 (最適状態系列)
最適な状態系列$q = (q_1 q_2 \ldots q_T)$の発見

- 観測系列Yがどの状態系列qから生成されたと考えられるか

$\Rightarrow$ 例：Viterbiアルゴリズム

問題3 (パラメータ推定)
$P(Y|\lambda)$を最大とするようなモデルパラメータλの調整

- 観測系列Yを生成するためのパラメータλの最適化

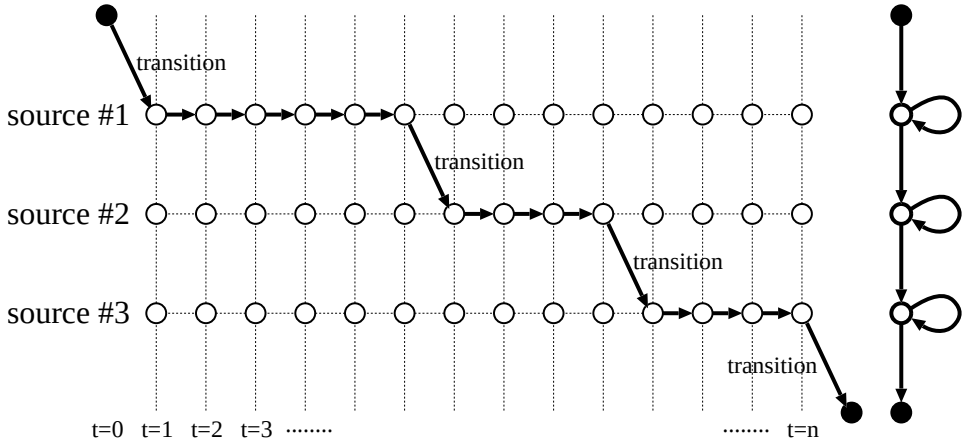
$\Rightarrow$ 例：Baum-Welchアルゴリズム

Andrew J. Viterbi

図2. A. J. Viterbi
1935年3月9日 イタリア生まれ

Viterbi 経路

- パラメータで規定されるモデルΛから、経路Qに沿って時系列$Y = \{y_1, y_2, \cdots, y_n\}$が生成される確率: $P(Y|Q, \Lambda)$
- どの時刻にどの信号源が活動するか、知ることができない
  → 確率最大になるように情報源の切り替えタイミングを調整
  → 組み合わせ爆発!! → DPの原理により、効率的に計算可能
- 例: $Q = \{1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3\}$



- Viterbi経路

$$\hat{Q} = \underset{Q}{\operatorname{argmax}}\, P(Y|Q, \Lambda)$$

- 単純マルコフ遷移を仮定

Viterbiアルゴリズム

Viterbiアルゴリズム

モデル λ が与えられた時、**観測信号系列** $Y = (y_1, y_2, ..., y_T)$ を出力する確率の最も高い**状態系列** $q = (q_1, q_2, \cdots, q_T)$ と、その状態系列のパスにおける確率を求める。

例

図のような確率を持つHMMを考える。

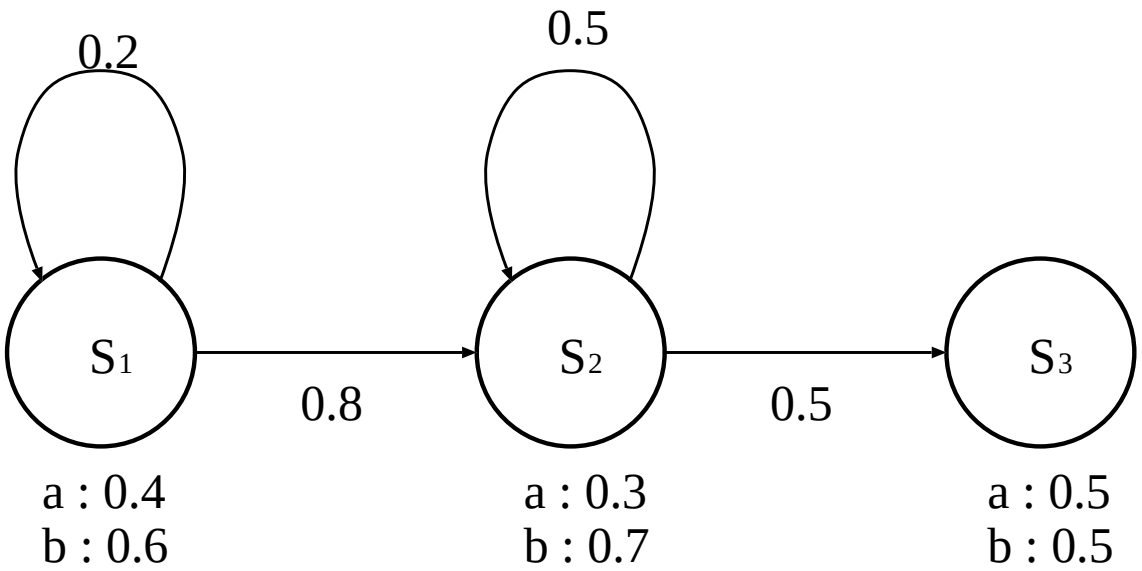


図3. HMMの例(left-to-right モデル)

Viterbiアルゴリズム

例

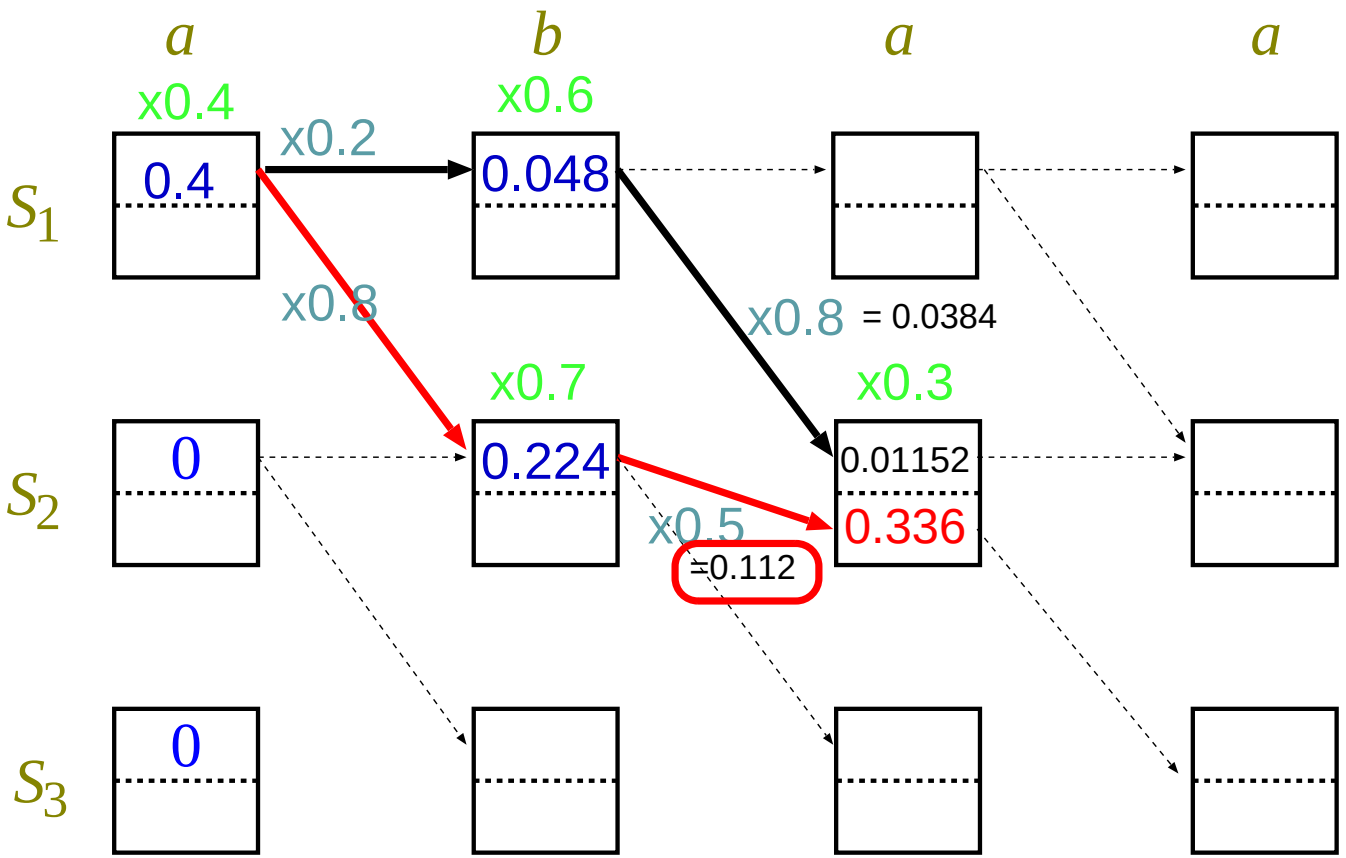


図4. Viterbiアルゴリズムの例

Viterbiアルゴリズム

例

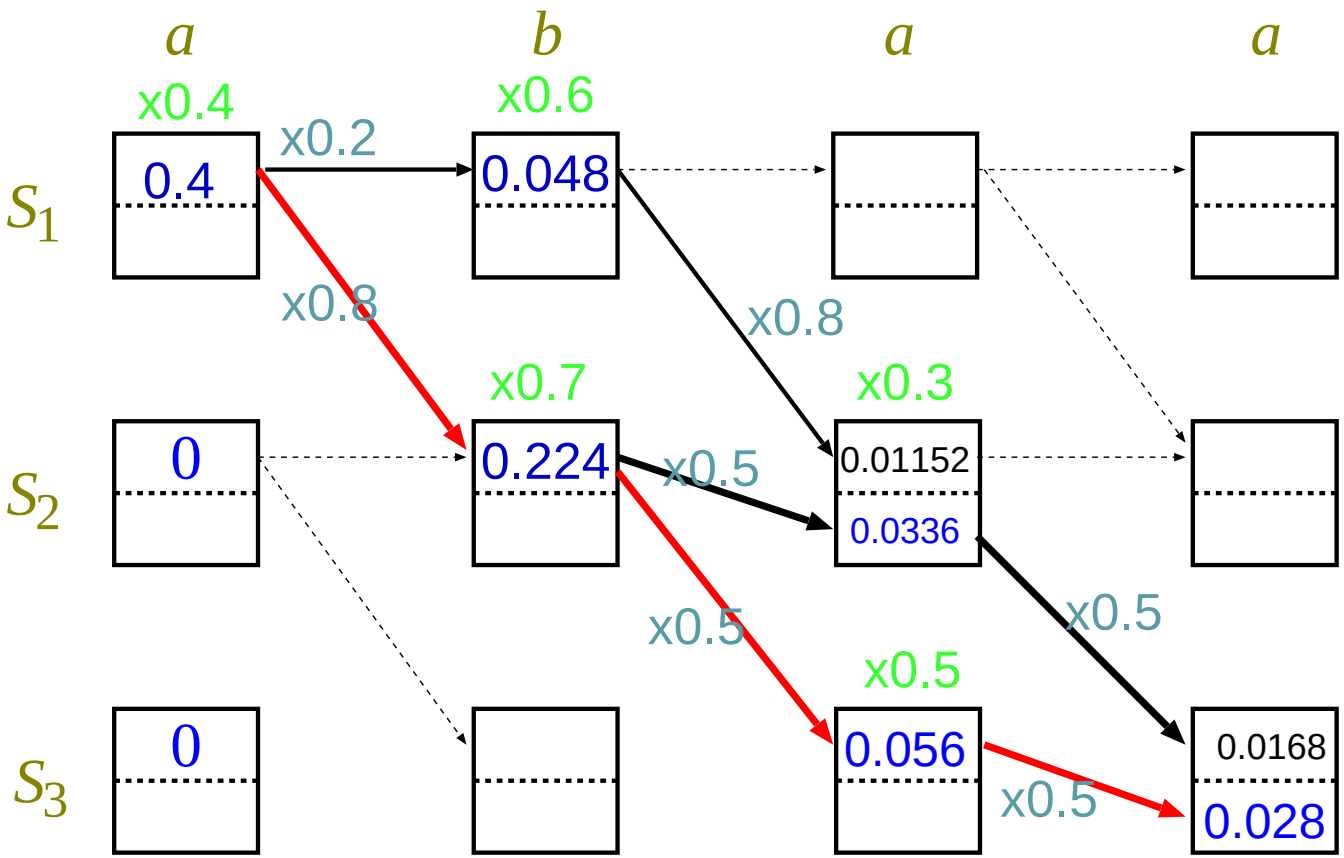


図5. Viterbiアルゴリズムの例

Viterbiアルゴリズム

観測系列 Y に対する最適な状態系列(尤度最大)を見つけるために、変数 $\delta_t(i)$ を(1)式のように、観測信号 y_t を出力して、かつ状態 q_i にある最大の確率と定義する。
この変数を(2)式のようにして再帰的に計算し、最も高い確率 P が得られる状態系列を求める。

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P[q_1 q_2 \ldots q_{t-1}, q_t = i, \boldsymbol{y}_1 \boldsymbol{y}_2 \ldots \boldsymbol{y}_t | \lambda] \tag{1}$$

$$\delta_{t+1}(j) = \left[\max_i \delta_t(i) a_{ij}\right] b_j(\boldsymbol{y}_{t+1}) \tag{2}$$

Viterbi アルゴリズム

Viterbi確率計算

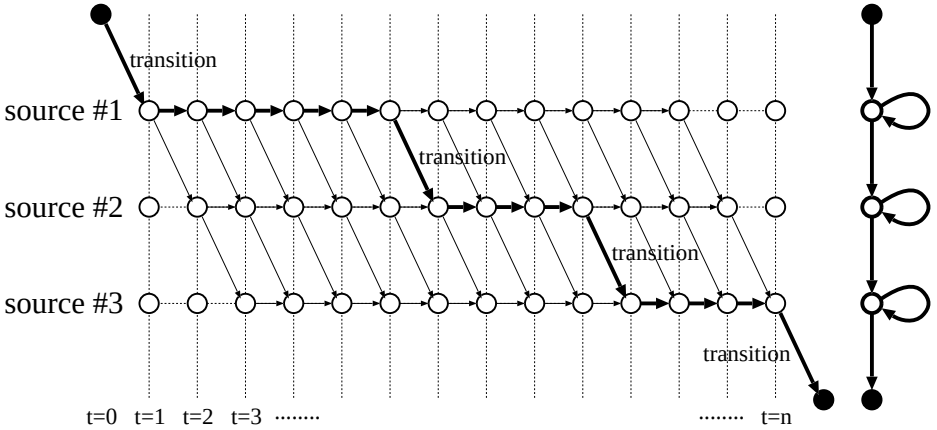
初期確率: $P_{0,i} = \pi_i$

漸化式:

$$P_{t,j} = \max_i \{\ a_{ij}\ b_i(\boldsymbol{y}_t)\ P_{t-1,i}\ \}$$

経路記録:

$$q_{t,j} = \operatorname{argmax}_i \{\ a_{ij}\ b_i(\boldsymbol{y}_t)\ P_{t-1,i}\ \}$$

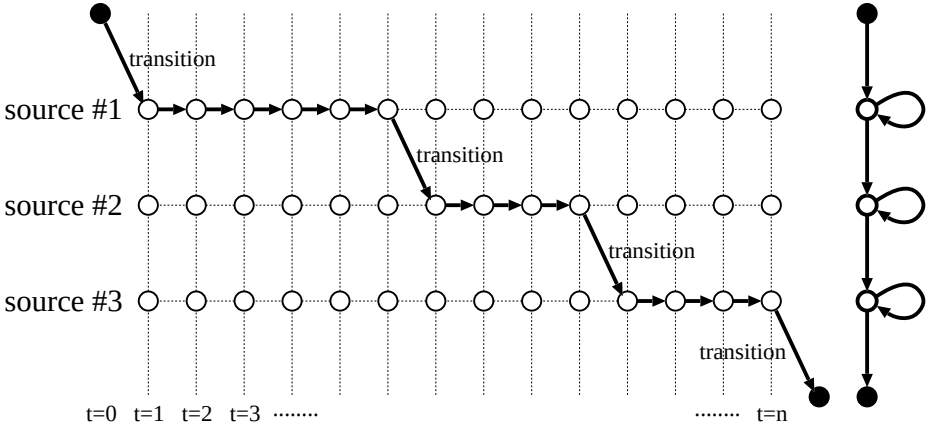


Viterbi経路決定 (traceback)

$$i \leftarrow q_{t+1,i} \quad t = T, \cdots, 2, 1$$

Viterbi アラインメント、
Viterbi セグメンテーション

例: $Q = \{1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3\}$



Viterbiアルゴリズム

■ 初期化

$$\delta_1(i) = \pi_i b_i(y_1)$$
$$\psi_1(i) = 0$$

■ 帰納

$$\delta_t(j) = \max_{i=1}^{N}[\delta_t - 1(i)a_{ji}]b_j(y_t)]$$

$$\psi_t(j) = \operatorname{argmax}_{i=1}^{N}[\delta_t - 1(i)a_{ji}]$$

■ 終了

$$P^* = \max_{i=1}^{S} \delta_T(i)$$

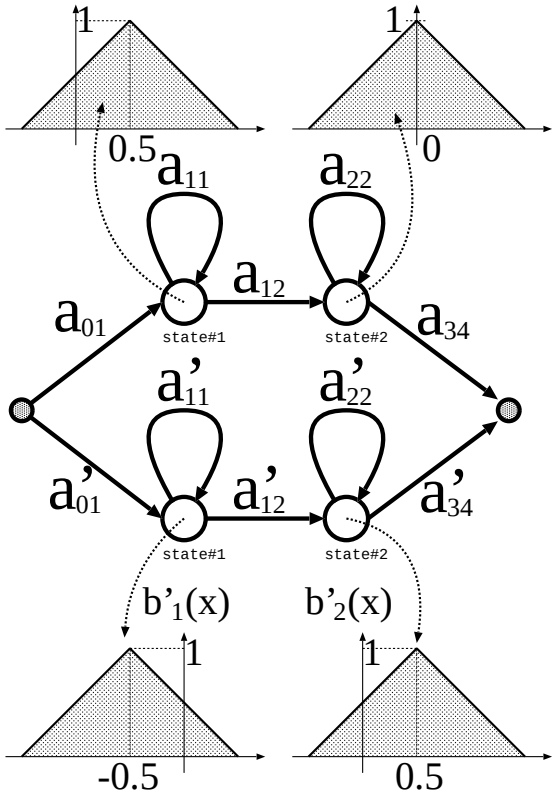
$$q_T^* = \operatorname{argmax}_{i=1}^{S} \delta_T(i)$$

■ トレースバック $t = T-1, T-2, \cdots, 1$

$$q_t^* = \psi_{t+1}(q_t + 1)$$

Viterbi アルゴリズムの演習 (過去問)

- ある対話の場面では、2つの単語「円」(eN) と「弧」(ko) のどちらかが発声されるが、「円」が発声される確率は「弧」の確率の4倍であることが分かっている。
- 入力音声信号を100mSごとにパワーを観測して、そのパターンにより簡単に音声認識を行おうと思う。(パワーだけで認識するなんてちょっと無茶苦茶だ。でもやってしまおう。)
- 観測値の単位は、dB値を10で割ったもの(つまりB(ベル))としよう。
- いずれの単語もスキップ無しの2状態のHMM(図1参照)でモデル化されている。出力確率密度は、図2のように3角形と仮定する。(これも少々乱暴だが、計算はしやすい。)
- 「円」に関して、状態1,2の停留確率はそれぞれ0.6, 0.7、出力確率密度はそれぞれ図のように平均値を0.5, 0 とする直角三角形とする。「弧」に関して、状態1,2,3の停留確率はそれぞれ0.4, 0.6、出力確率密度はそれぞれ図のように平均値を-0.5, 0.5 とする直角三角形とする。
- さて、観測されたパワーのデータは、$\{-0.2, 0.1, 0.4, 0.3\}$ であった。下の「弧」に関する例を参考に、「円」のTrellisの図を描き、2 桁精度程度のViterbi計算を行い、の図中にViterbi経路を描きなさい。この発声は、どちらの単語か音声認識結果を答えなさい。



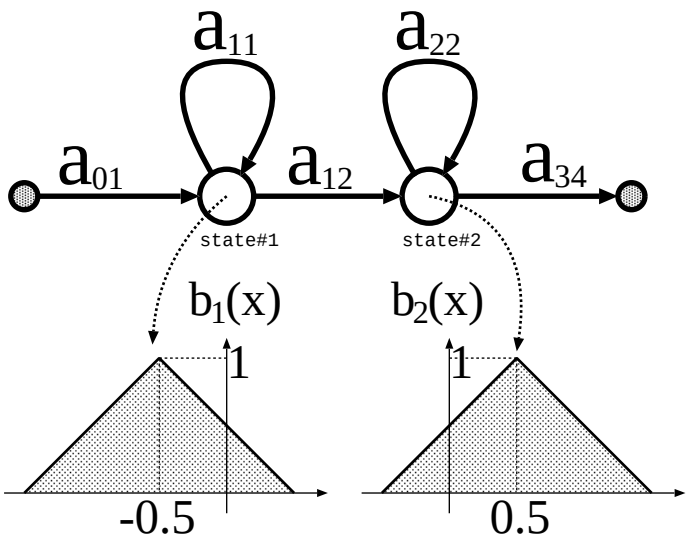
Viterbi アルゴリズムの演習

■ **モデルパラメータ: Λ**

$$a_{01} = 0.2,$$
$$a_{11} = 0.4,\ a_{12} = 0.6,$$
$$a_{22} = 0.6,\ a_{23} = 0.4,$$
$$b_1(x) = \max(1 - |x + 0.5|, 0),$$
$$b_2(x) = \max(1 - |x - 0.5|, 0)$$

■ **観測データ: X =**

$$\{\ -0.2,\ 0.1,\ 0.4,\ 0.3\ \}$$



単語/ko/「弧」の trellis 図と Viterbi 経路

時刻 t 観測値x(t)		初期 →	1 -0.2 →	2 0.1 →	3 0.4 →	4 0.3 →	終り
状態	確率	1.0					
	遷移 0.2		0.2	(0.7)	(0.4)	(0.1)	(0.2)	
1	停留 0.4	0 → 0 →	0.14 →	0.056 →	0.022 →	0.0088 →	0.00088 →	→ →
	遷移 0.6		0	(0.3) →	0.084	(0.6) →	0.013	(0.9) →	0.00053	(0.8)
2	停留 0.6	0 → 0 →	0 → 0 →		0.050 →	0.030 →	0.027 →	0.016 →	0.0013 →	
	遷移 0.4							0.00052

Viterbi アルゴリズムのCプログラム例

```
/* 構造体 STATE, 最大データ長 N, 最大次元 P, 最大状態数 S, 対数尤度 Loglikelihood() は定義されているとする */
double ViterbiPath(     /* Viterbi score を返す */
    int n,              /* データ長 */
    double data[N][P],  /* P次元ベクトル列、長さnのデータ data[0..n-1][0..P] */
    int s,              /* 隠れ状態数 */
    STATE state[S],     /* s個の隠れ状態構造体の配列 */
    int segm[S])        /* 出力:  Viterbi セグメンテーション; segm[0..s-2] */
{
    char path[N][S];
    double vec[P],like[S],stay,move,score,log(),LogLikelihood();
    int i,j,k,l,t;

    /* Viterbi アルゴリズムによる最適経路の探索と対数尤度の積算 */
    like[0]=LogLikelihood(data[0],&state[0]);
    for(i=1;i<s;i++) like[i]= -1000000; /* -∞ をセット */
    for(t=1;t<n;t++) { /* 時刻ごとに Viterbi 演算を進行させる */
        for(i=s-1;i>0;i--) {
            stay=like[i]+state[i].logstay;
            move=like[i-1]+state[i-1].logtran;
            if(stay > move) { like[i]=stay; path[t][i]=0; }
            else            { like[i]=move; path[t][i]=1; }
            like[i]+=LogLikelihood(data[t],&state[i]);
        }
        like[0]+=state[0].logstay+LogLikelihood(data[t],&state[0]);
        path[t][0]=0;
    }
    score=like[s-1]+state[s-1].logtran; /* 最後の遷移 */

    /* Viterbi トレースバック */
    segm[j=s-1]=n; for(i=n-1; i>0&&j>=0; i--) if(path[i][j]) segm[--j]=i;
    return(score);
}
```

Viterbi経路、Viterbiセグメンテーション

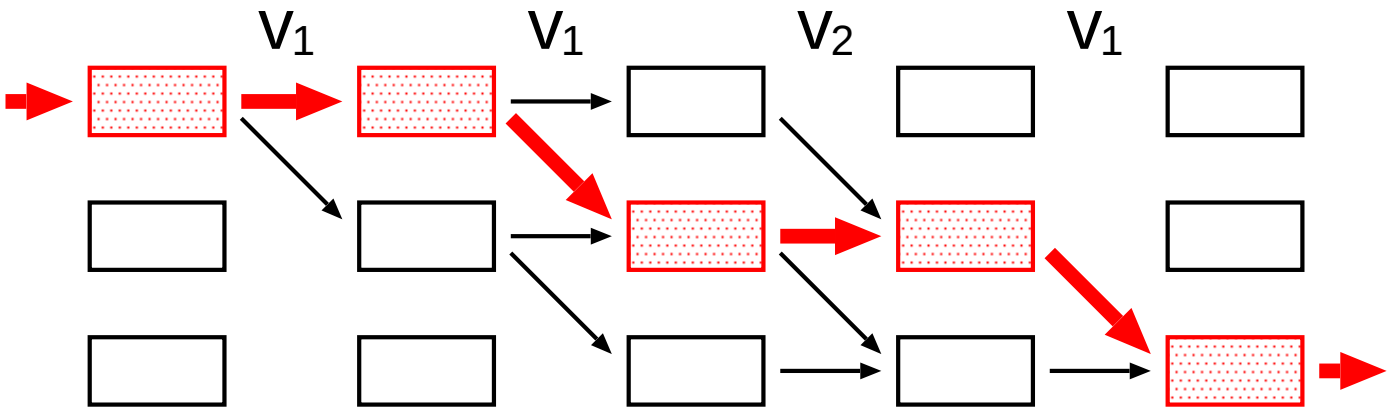


図6. Viterbi経路

Viterbiアルゴリズムによって得られる経路を
Viterbi経路と呼ぶ

⇓

尤度が最大になるように各状態ごとに状態系列を分割
(Viterbiセグメンテーション)

Viterbiアルゴリズムと Forwardアルゴリズム

■ 最大確率 vs 確率総和

$$P_{\text{Viterbi}} = \max_{Q} P(Y|Q) \qquad \textbf{vs} \qquad P_{\text{Forward}} = \sum_{Q} P(Y|Q)$$

■ 確率計算アルゴリズム

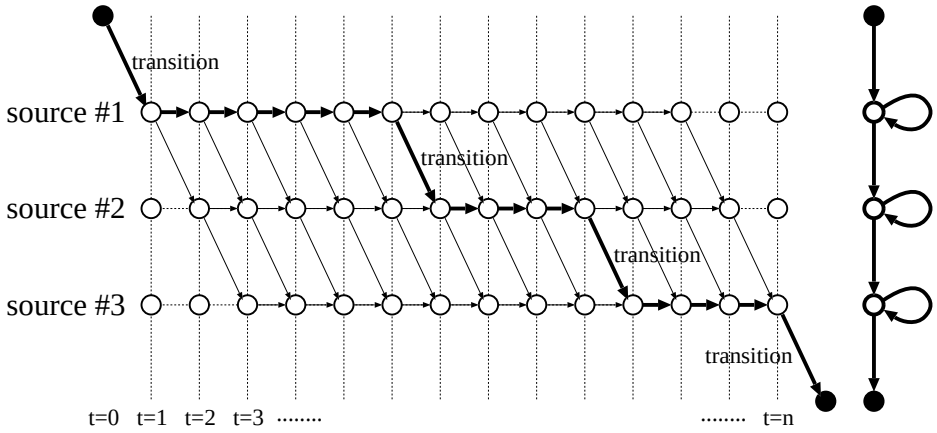
初期確率: $P_{0,i} = \pi_i$

Viterbi確率計算漸化式:

$$P_{t,j} = \max_{i}\{\ a_{ij}\ b_i(\boldsymbol{y}_t)\ P_{t-1,i}\ \}$$

Forward確率計算漸化式:

$$P_{t,j} = \sum_{i}\{\ a_{ij}\ b_i(\boldsymbol{y}_t)\ P_{t-1,i}\ \}$$



単語HMMを用いた単語音声認識

認識アルゴリズム

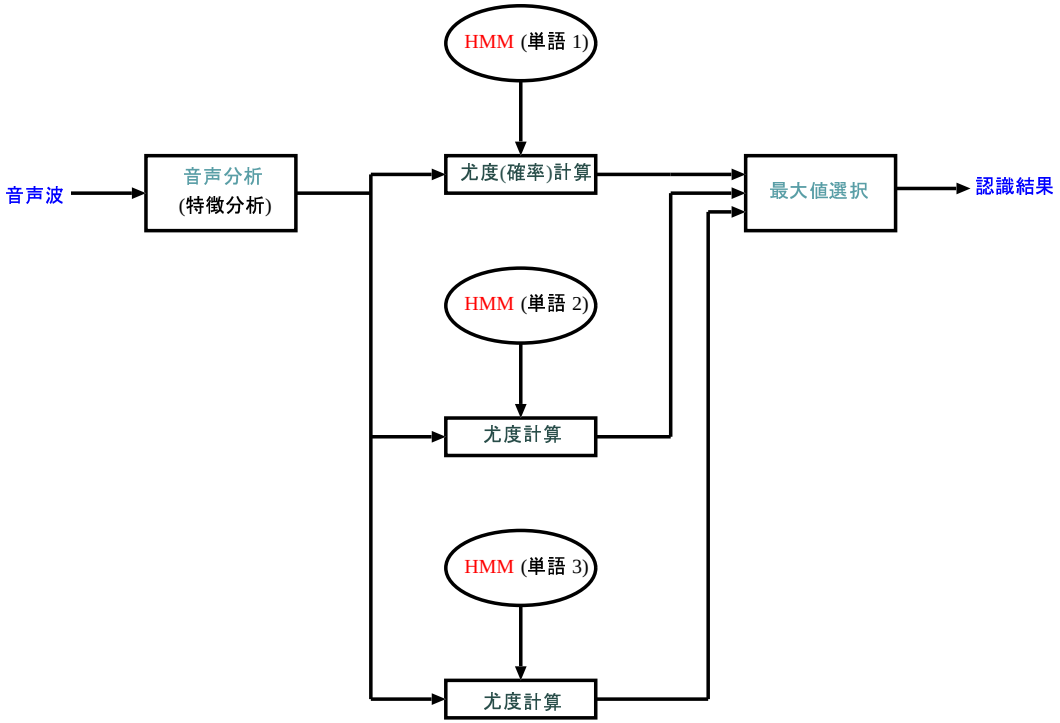
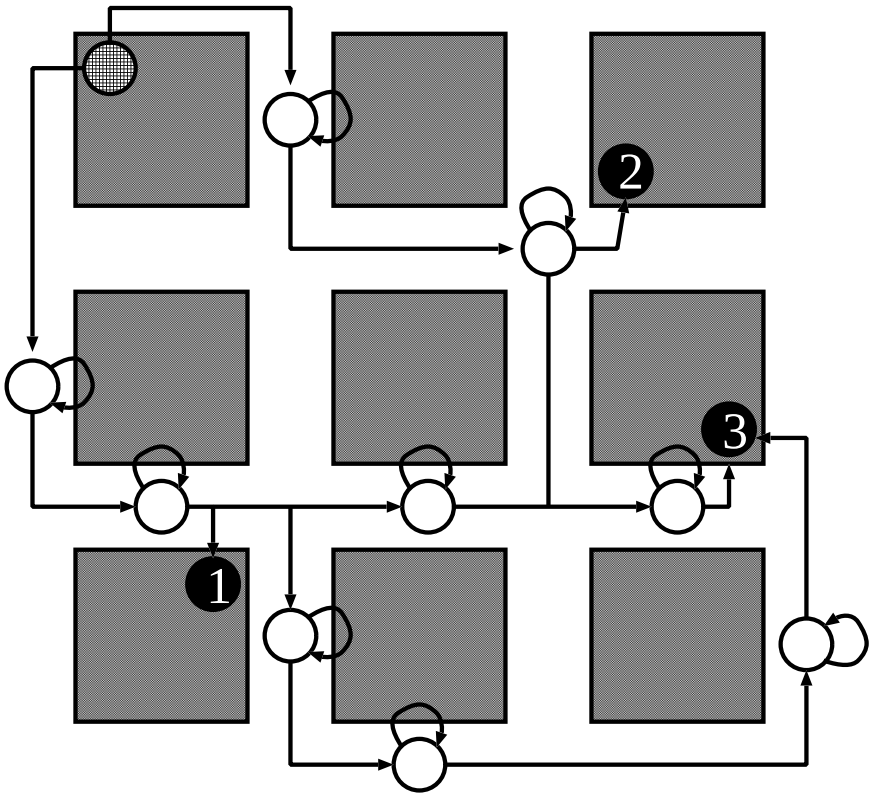


図7. 単語HMMを用いた単語音声認識の方法

各HMMモデルごとに、
観測系列$Y = (y_1 y_2 \cdots y_T)$が生起する確率（尤度）$P(Y|\lambda)$を求め、
最大確率（最大尤度）を与えるモデルを選んで、認識結果とする。

時系列認識 — 音声認識の原理

■ **シャーロック・ホームズ「ギリシア語通訳事件」: 目隠しをされて馬車に乗せられて、ある秘密の場所に拉致された。時々刻々聞こえて来る音情報y_tの列から、それらしい場所を突き止める問題。**(市場の近くを通ればその賑わいが聞こえ、路面が工事中ならば車輪の音が変化して聞こえてくる。乗せられた場所から、可能性のある目的地までの経路すべてをW_iと表して、順次聞こえて来た音をうまく説明できる経路はどれか、その確率を考える。)



■ **経路仮説Wの確率:** $P(W)$
仮説Wから音Yを聞く確率: $P(Y|W)$
問題: $\underset{W}{\operatorname{argmax}}\, P(W|Y)$

HMMの音声認識への応用

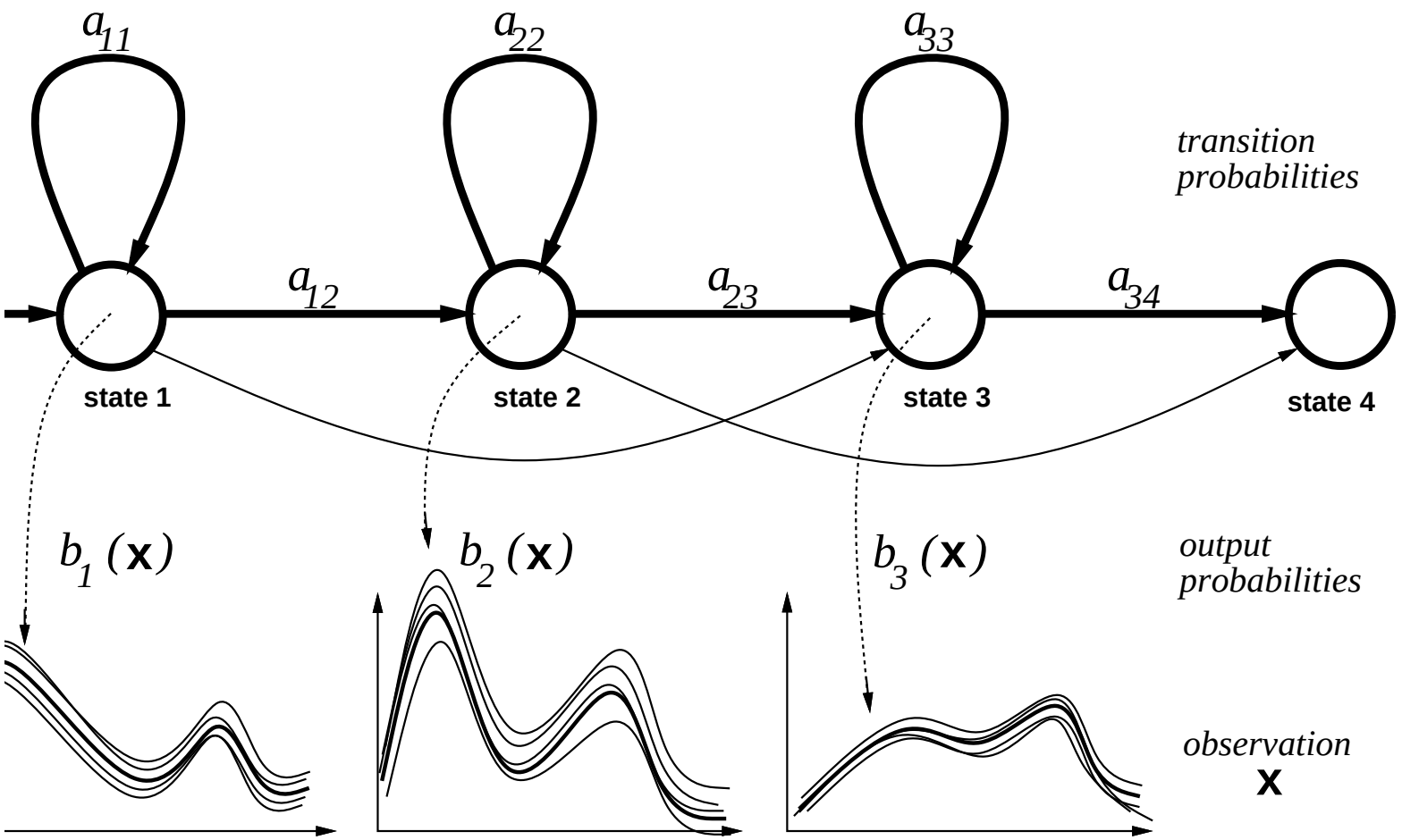


図8. 音声認識のためのHMMの意味

Viterbi学習

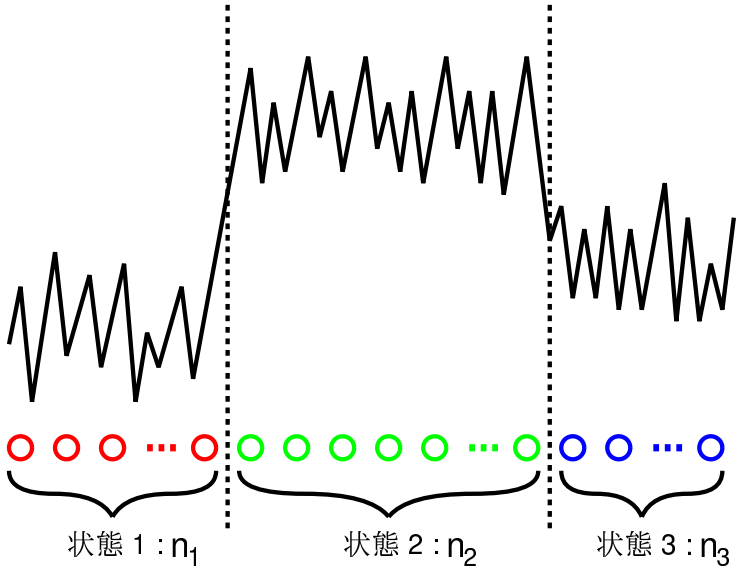


図9. Viterbiセグメンテーション

Viterbiセグメンテーションにより観測状態系列を分割
⇔ 最尤推定によりモデルパラメータλを修正

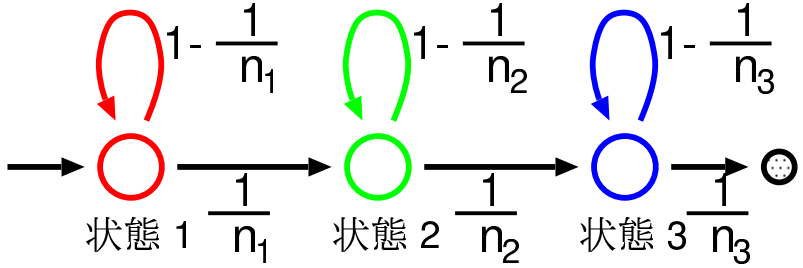


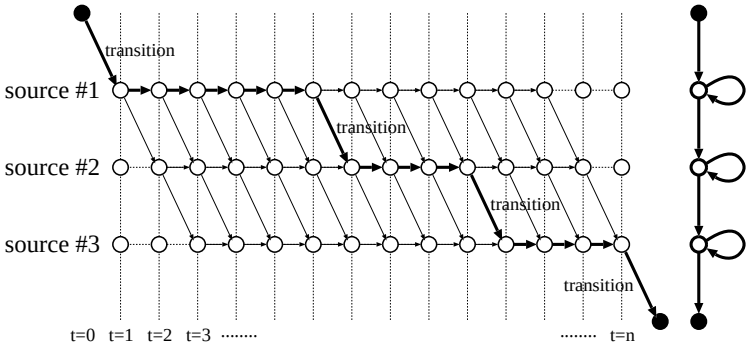
図10. HMMモデル

Viterbi 学習

- **モデルパラメータ $\Lambda = \{a_{ij}, b_i(\boldsymbol{y}_t)\}$、時系列 $Y = \{\boldsymbol{y}_t\}$**
- **二段階の最適化: 収束保証**
  1. **Λ の初期区分化: Segmental k-means clustering, 単純等分割など**
  2. **Viterbi 経路を求める:**
     経路 Q について確率最大化

$$\hat{Q} = \underset{Q}{\operatorname{argmax}}\, P(Y|Q, \Lambda)$$

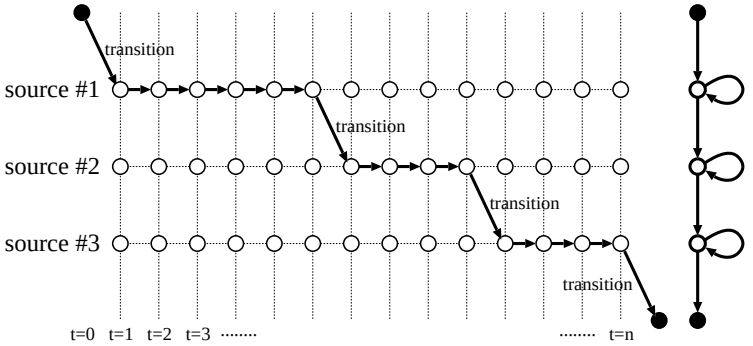
例: $Q = \{1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3\}$



  3. **Viterbi 経路から Λ を最尤推定:**
     Λ について確率最大化

$$\hat{\Lambda} = \underset{\Lambda}{\operatorname{argmax}}\, P(Y|\hat{Q}, \Lambda)$$

単一正規分布: $\mu \leftarrow$ 標本平均、$\sigma^2 \leftarrow$ 標本分散



  4. **繰り返し: $\Lambda \leftarrow \hat{\Lambda}, Q \leftarrow \hat{Q}$ (収束するまで)**
- **複数の時系列 $\{Y_i\}$: 全系列の対応区間から Λ を最尤推定**