HMM:隠れマルコフモデル

電子情報工学科
伊庭　斉志

マルコフモデルと隠れマルコフモデル

- 一次マルコフ連鎖
  - 状態集合

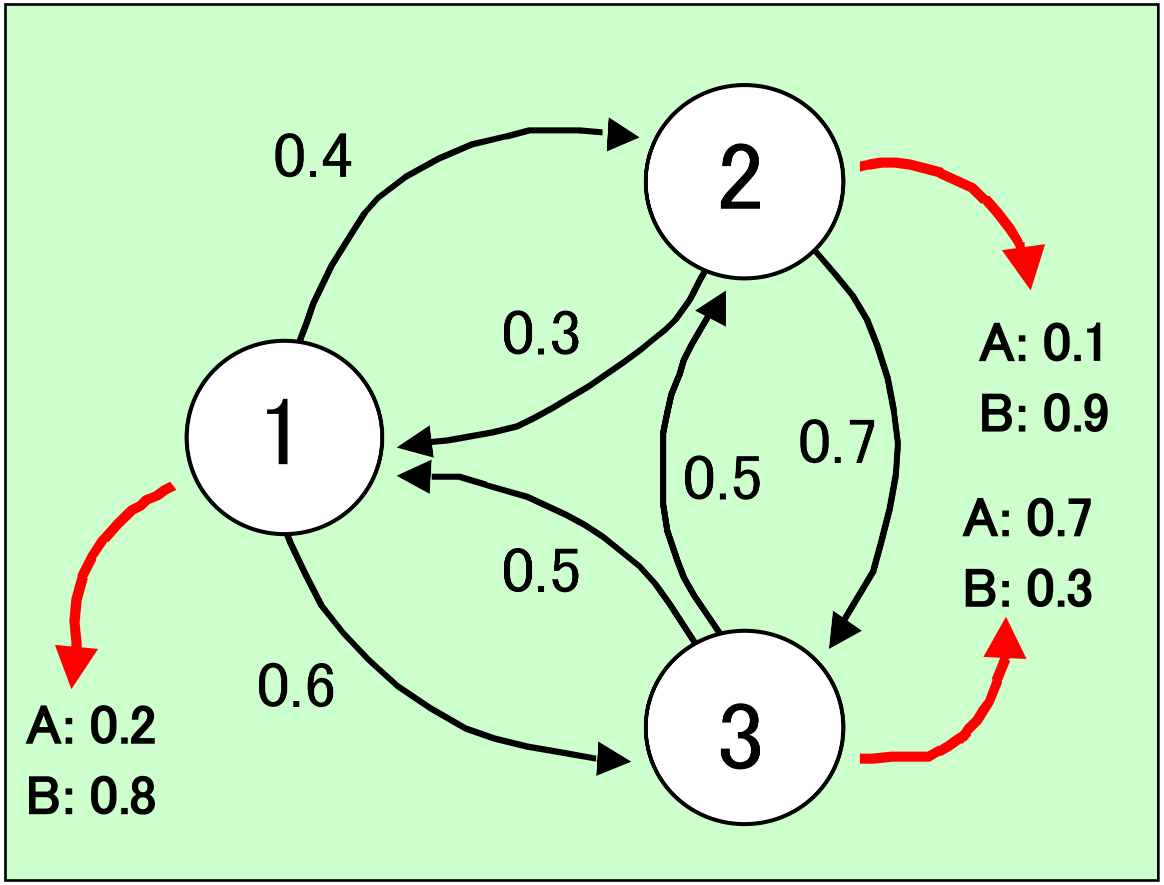
    $S=\{1,2,\ldots n\}$
  - 遷移確率($k \rightarrow l$)

    $$a_{kl}$$
- 隠れマルコフモデル（Hidden MM:HMM）
  - 出力記号集合Σ
  - 出力確率(状態から出力記号への写像)

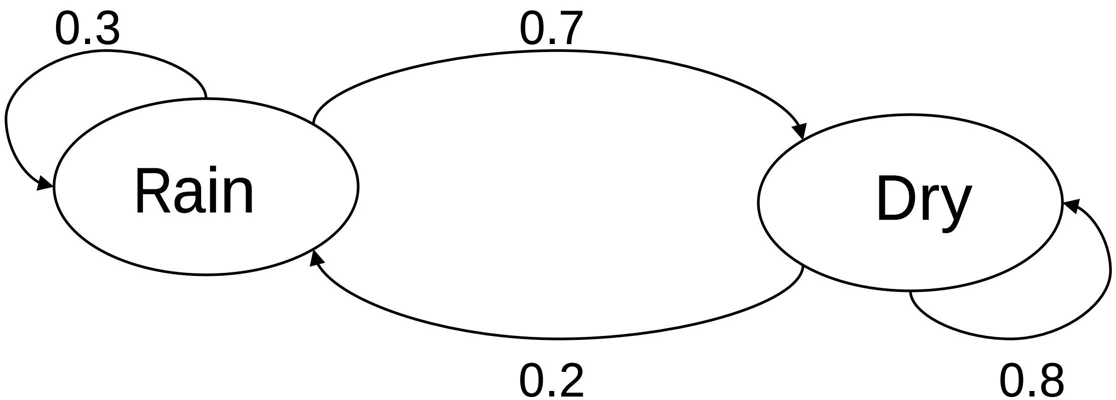
    $$e_k(b) : S \rightarrow \Sigma$$

隠れマルコフモデル(HMM)

- HMM≒有限オートマトン＋確率
- 定義
  - 出力記号集合Σ
  - 状態集合
    $S=\{1,2,\ldots n\}$
  - 遷移確率($k \rightarrow l$)
    $$a_{kl}$$
  - 出力確率
    $$e_k(b)$$
  - 開始状態
    終了状態



マルコフモデルの例

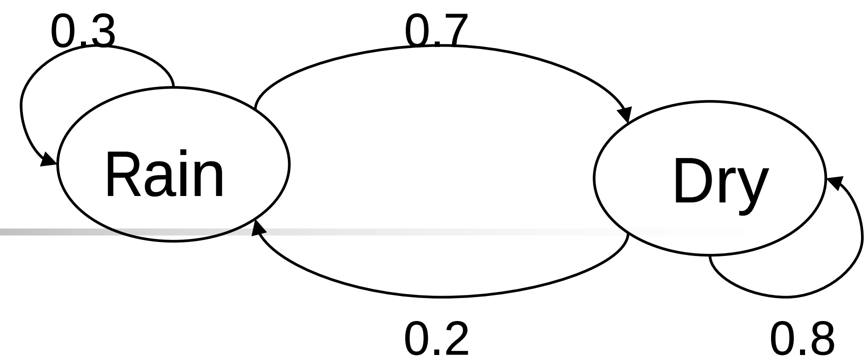


- ２つの状態: ‘Rain’ と ‘Dry’.
- 推移確率:

  P(‘Rain’|‘Rain’)=0.3 , P(‘Dry’|‘Rain’)=0.7 ,

  P(‘Rain’|‘Dry’)=0.2, P(‘Dry’|‘Dry’)=0.8
- 初期確率: P(‘Rain’)=0.4 , P(‘Dry’)=0.6 .

計算例



- マルコフ性から

$$P(s_{i1}, s_{i2}, \ldots, s_{ik}) = P(s_{ik} \mid s_{i1}, s_{i2}, \ldots, s_{ik-1}) P(s_{i1}, s_{i2}, \ldots, s_{ik-1})$$

$$= P(s_{ik} \mid s_{ik-1}) P(s_{i1}, s_{i2}, \ldots, s_{ik-1}) = \ldots$$

$$= P(s_{ik} \mid s_{ik-1}) P(s_{ik-1} \mid s_{ik-2}) \ldots P(s_{i2} \mid s_{i1}) P(s_{i1})$$

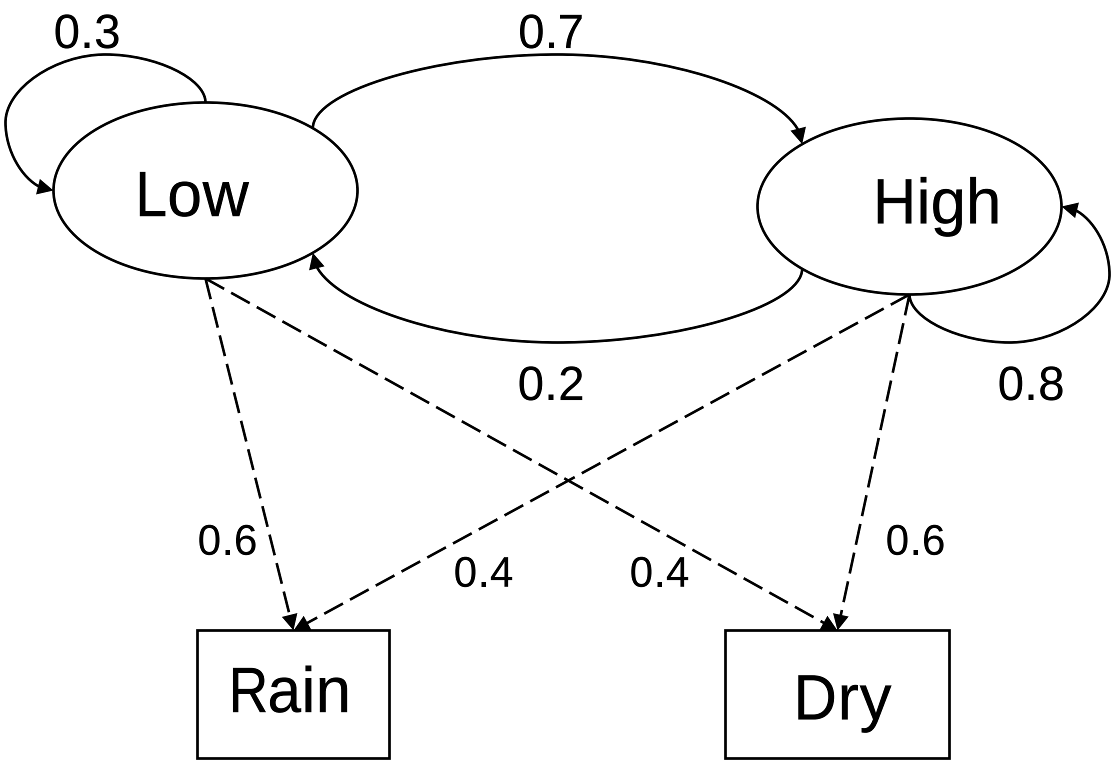
- 例えば、{'Dry','Dry','Rain',Rain'}の状態列の確率は、

P({'Dry','Dry','Rain',Rain'}) =

P('Rain'|'Rain') P('Rain'|'Dry') P('Dry'|'Dry') P('Dry')=

= 0.3*0.2*0.8*0.6

隠れマルコフモデルの例



隠れマルコフモデルの例

- 2つの状態: ‘Low’ と ‘High’（気圧）
- 2つの観測：‘Rain’ と‘Dry’.
- 推移確率:

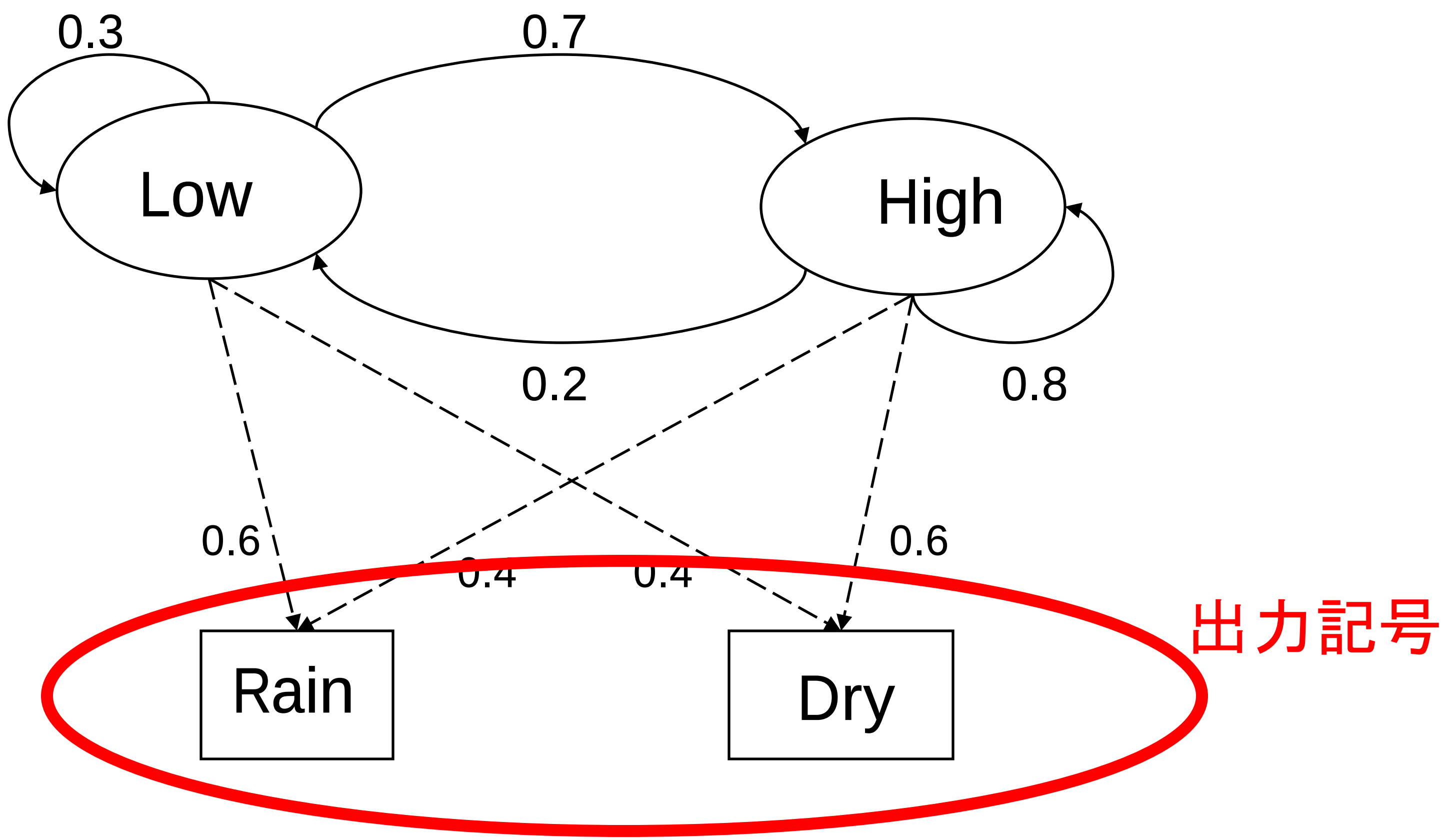
  P(‘Low’|‘Low’)=0.3 , P(‘High’|‘Low’)=0.7 ,

  P(‘Low’|‘High’)=0.2, P(‘High’|‘High’)=0.8
- 観測確率：

  P(‘Rain’|‘Low’)=0.6 , P(‘Dry’|‘Low’)=0.4 ,

  P(‘Rain’|‘High’)=0.4 , P(‘Dry’|‘High’)=0.3 .
- 初期確率: P(‘Low’)=0.4 , P(‘High’)=0.6 .

隠れマルコフモデルの例



計算例

- {‘Dry’,’Rain’}のような観測列がえられる確率は？
- すべての可能な隠れ状態列の確率を求める

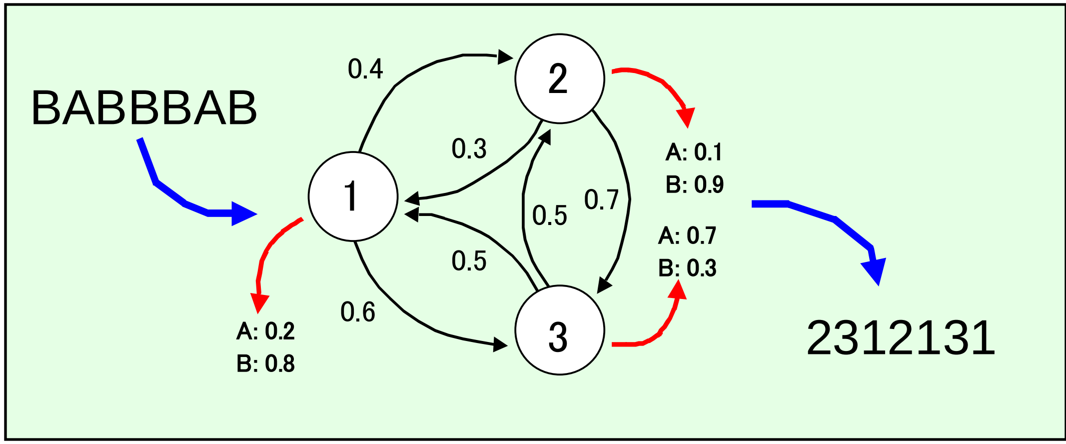
P({‘Dry’,’Rain’}) =
P({‘Dry’,’Rain’} , {‘Low’,’Low’}) +
P({‘Dry’,’Rain’} , {‘Low’,’High’}) +
P({‘Dry’,’Rain’} , {‘High’,’Low’}) +
P({‘Dry’,’Rain’} , {‘High’,’High’})

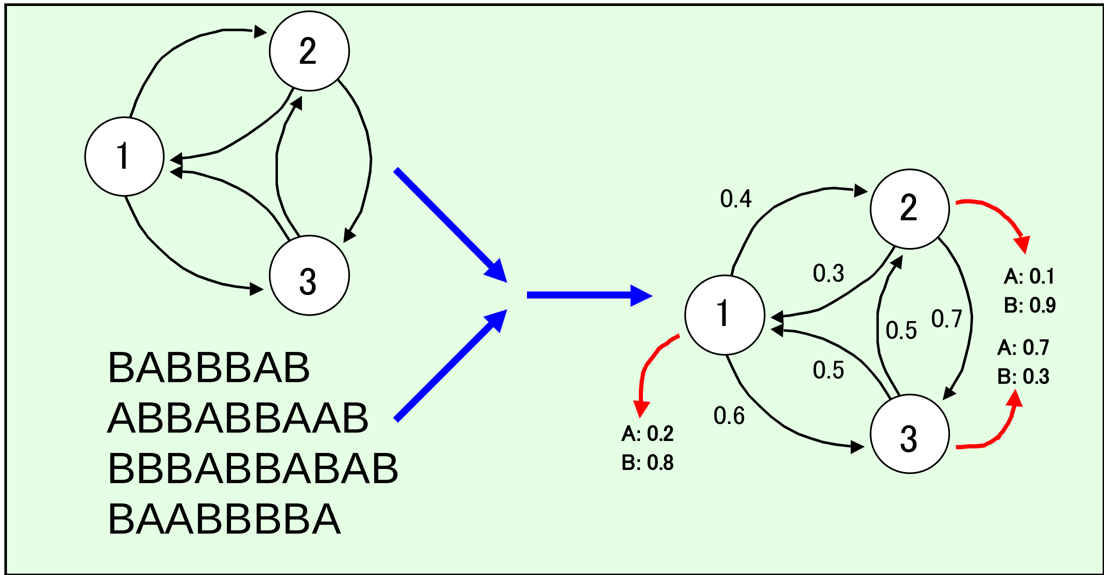
ただし:

P({‘Dry’,’Rain’} , {‘Low’,’Low’})=
P({‘Dry’,’Rain’} | {‘Low’,’Low’}) P({‘Low’,’Low’}) =
P(‘Dry’|’Low’)P(‘Rain’|’Low’) P(‘Low’)P(‘Low’|’Low)
= 0.4*0.4*0.6*0.4*0.3

HMMにおける基本アルゴリズム

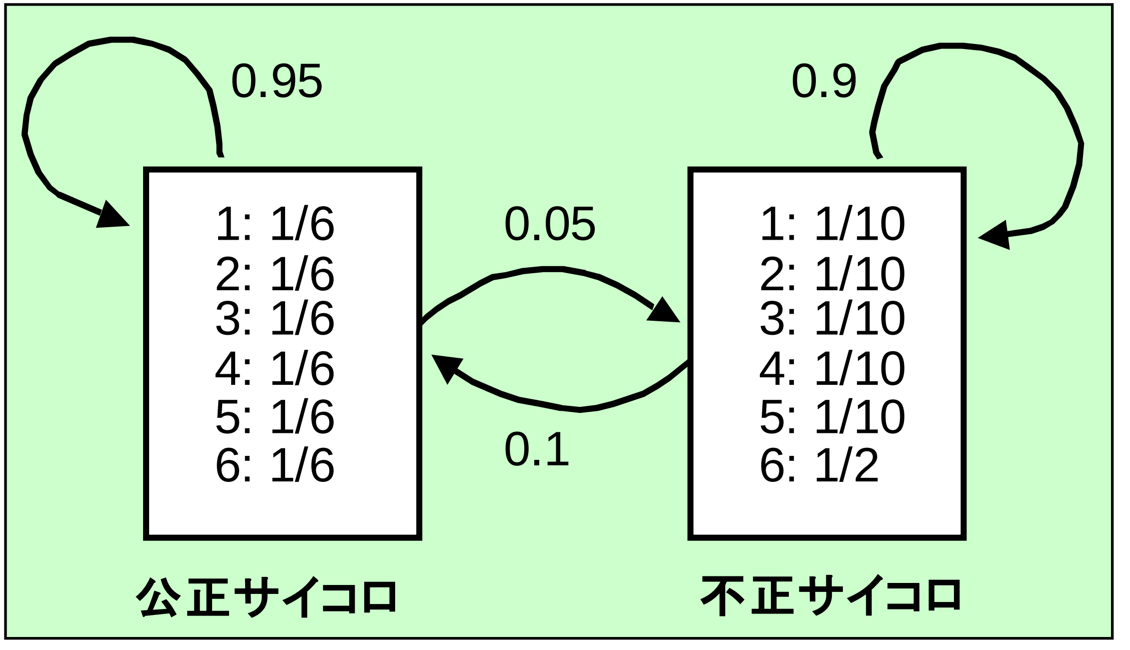
- Viterbiアルゴリズム
  - 出力記号列から状態列を推定
  - 構文解析
- Baum-Welchアルゴリズム
  (EMアルゴリズム)
  - 出力記号列からパラメータを推定
  - 学習





時々いかさまをするカジノ

- サイコロの出目だけが観測可能、どちらのサイコロを振っているかは観測不可能
- サイコロの出目から、どちらのサイコロを振っているかを推定
- 6,2,6,6,3,6,6,6,
  4,6,5,3,6,6,1,2
  →不正サイコロ
- 6,1,5,3,2,4,6,3,
  2,2,5,4,1,6,3,4
  →公正サイコロ
- 6,6,3,6,5,6,6,1,
  5,4,2,3,6,1,5,2
  →途中で公正サイコロに交換

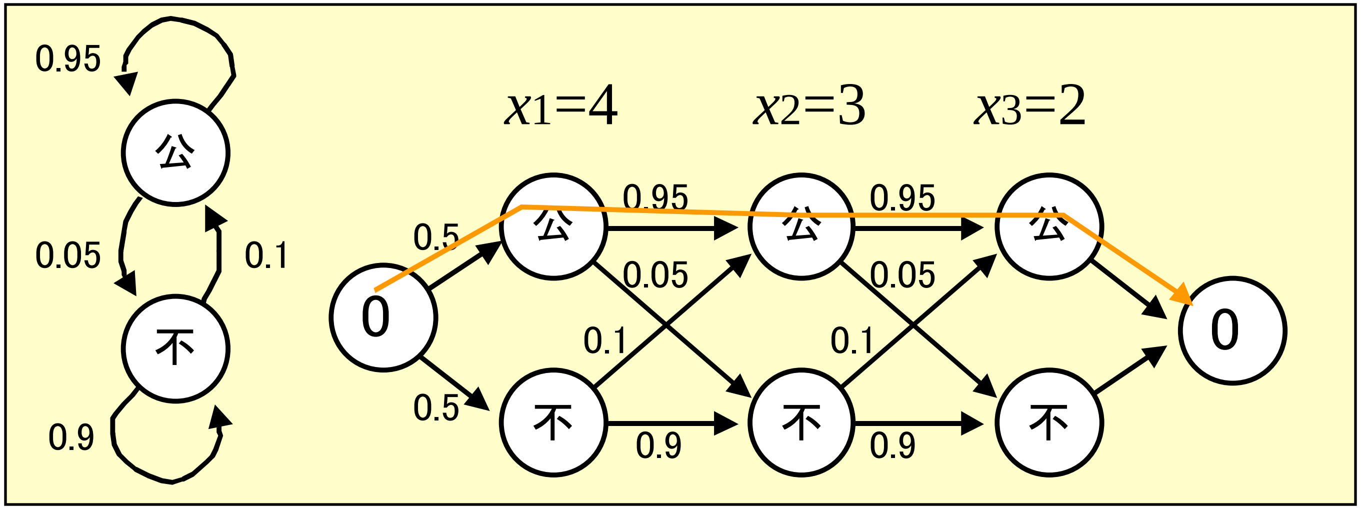


Viterbi アルゴリズム(1)

- 観測列(出力配列データ) $x=x_1\ldots x_L$と状態列$\pi=\pi_1\ldots\pi_L$が与えられた時、その同時確率は

$$P(x,\pi)=a_{0\,\pi_1}\Pi e_{\pi i}(x_i)a_{\pi i\pi i+1} \qquad \text{但し、}\pi_{L+1}=0$$

- x が与えられた時、最も尤もらしい状態列は $\pi^*=\mathrm{argmax}_\pi\ P(x,\pi)$
- 例:どちらのサイコロがいつ使われたかを推定



$$\max_\pi P(x_1x_2x_3,\pi)=P(x_1x_2x_3,\text{公公公})=0.5\cdot\tfrac{1}{6}\cdot 0.95\cdot\tfrac{1}{6}\cdot 0.95\cdot\tfrac{1}{6}$$

Viterbi アルゴリズム(2)

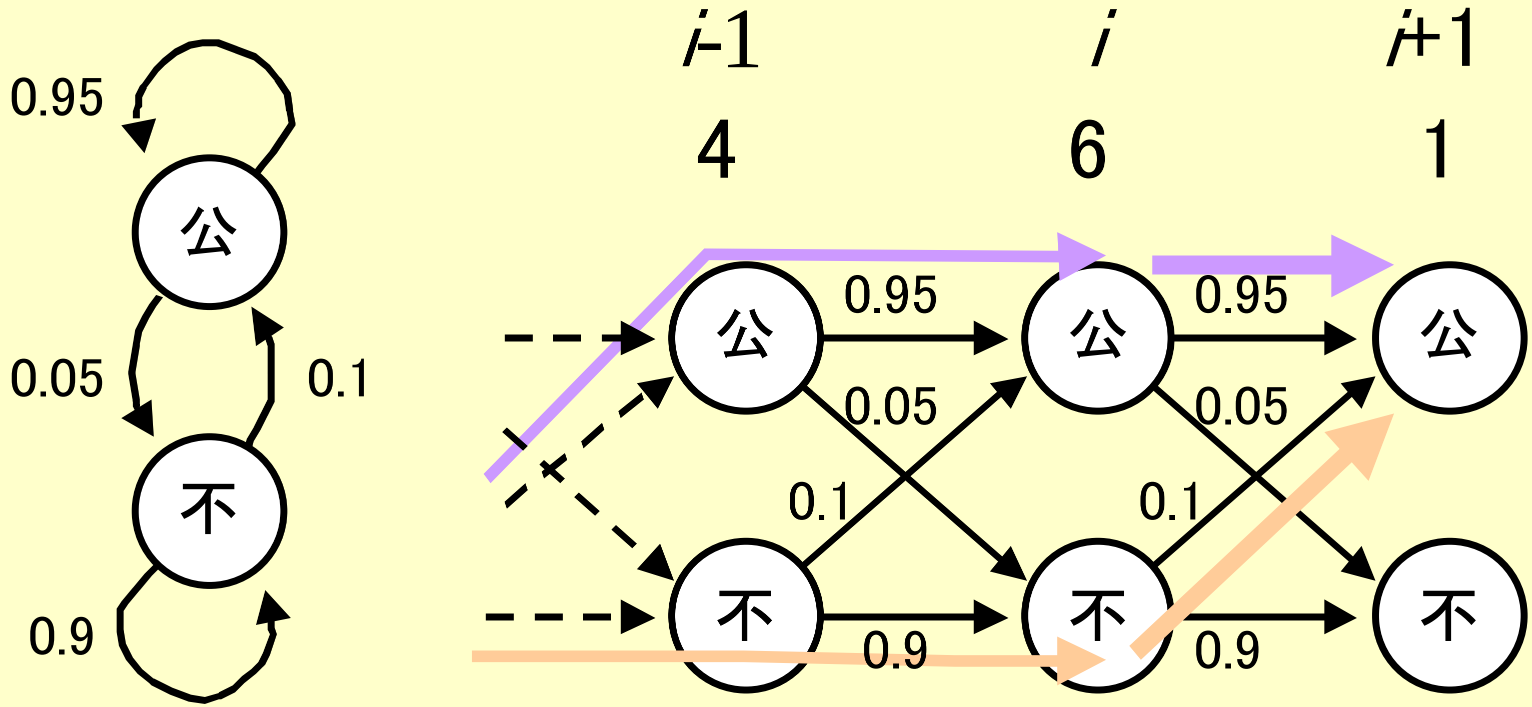
- x から、$\pi^*=\mathrm{argmax}_\pi\ P(x,\pi)$ を計算
- そのためには $x_1 \ldots x_i$ を出力し、状態 k に至る確率最大の状態列の確率 $v_k(i)$ を計算

$$v_k(i) = \max_\pi a_{0\pi_1} \prod_{j=1}^{i-1} e_{\pi_j}(x_j) a_{\pi_j \pi_{j+1}}$$

- $v_k(i)$は以下の式に基づき動的計画法で計算

$$v_l(i+1) = e_l(x_{i+1}) \max_k (v_k(i) a_{kl})$$

Viterbi アルゴリズム(3)



$$v_{公}(i+1) = \max\{\ e_{公}(1)\cdot 0.95\cdot v_{公}(i),\ e_{公}(1)\cdot 0.1\cdot v_{不}(i)\ \}$$

EM(Expectation Maximization)アルゴリズム

- 「欠けているデータ」のある場合の最尤推定のための一般的アルゴリズム

x:観測データ、y:欠けているデータ、

θ:パラメータ集合

目標：$\log P(x|\theta) = \log \sum_{y} P(x,y|\theta)$ の最大化

- 最大化は困難であるので、反復により尤度を単調増加させる(θ^tよりθ^{t+1}を計算)
- HMMの場合、「欠けているデータ」は状態列

EM(Expectation Maximization) アルゴリズムの導出

$$\log P(x\mid\theta) = \log P(x,y\mid\theta) - \log P(y\mid x,\theta)$$

両辺に$P(y\mid x,\theta^t)$をかけてyについての和をとり、

$$\log P(x\mid\theta) = \sum_y P(y\mid x,\theta^t)\log P(x,y\mid\theta) - \sum_y P(y\mid x,\theta^t)\log P(y\mid x,\theta)$$

右辺第１項を$Q(\theta\mid\theta^t)$とおくと、

$$\log P(x\mid\theta) - \log P(x\mid\theta^t) =$$

$$Q(\theta\mid\theta^t) - Q(\theta^t\mid\theta^t) + \sum_y P(y\mid x,\theta^t)\log\frac{P(y\mid x,\theta^t)}{P(y\mid x,\theta)}$$

最後の項は相対エントロピーで常に正なので、

$$\log P(x\mid\theta) - \log P(x\mid\theta^t) \geq Q(\theta\mid\theta^t) - Q(\theta^t\mid\theta^t)$$

よって、$\theta^{t+1} = \arg\max_\theta Q(\theta\mid\theta^t)$ とすれば尤度は増大

EM(Expectation Maximization) アルゴリズムの一般形

1. 初期パラメータ Θ^0 を決定。t=0とする
2. $Q(\theta|\theta^t)=\sum P(y|x,\theta^t)\log P(x,y|\theta)$ を計算
3. $Q(\theta|\theta^t)$を最大化するθ^*を計算し、 $\theta^{t+1}=\theta^*$ とする。$t=t+1$とする
4. Qが増大しなくなるまで、2，3を繰り返す

EM algorithm

- Start with initial estimate, θ^0

- Repeat until convergence
  - E-step: calculate

$$Q(\theta,\theta^t) = \sum_{i=1}^{n}\sum_{y} P(y \mid x_i,\theta^t)\log P(x_i, y \mid \theta)$$

  - M-step: find

$$\theta^{(t+1)} = \arg\max_{\theta} Q(\theta,\theta^t)$$

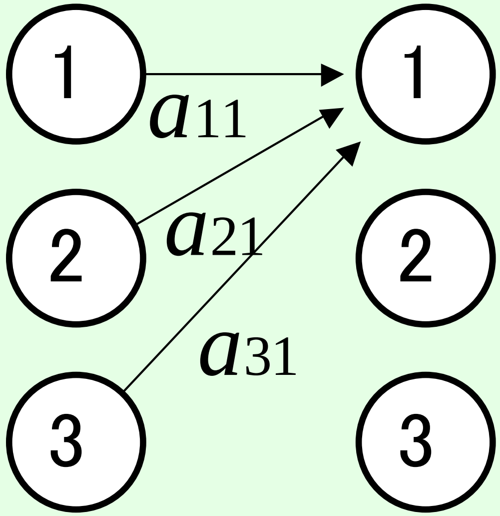
前向きアルゴリズム

- 配列 x の生成確率 $P(x)=\sum P(x,\pi)$ を計算
- Viterbiアルゴリズムと類似
- $f_k(i)=P(x_1\ldots x_i,\pi_i=k)$ をDPにより計算

$$f_0(0)=1,\ f_k(0)=0$$

$$f_l(i)=e_l(x_i)\sum_k f_k(i-1)a_{kl}$$

$$P(x)=\sum_k f_k(L)a_{k0}$$



$$f_1(i)=e_1(x_i)\,(f_1(i-1)a_{11}+f_2(i-1)a_{21}+f_3(i-1)a_{31})$$

Viterbiと前向きアルゴリズムの比較

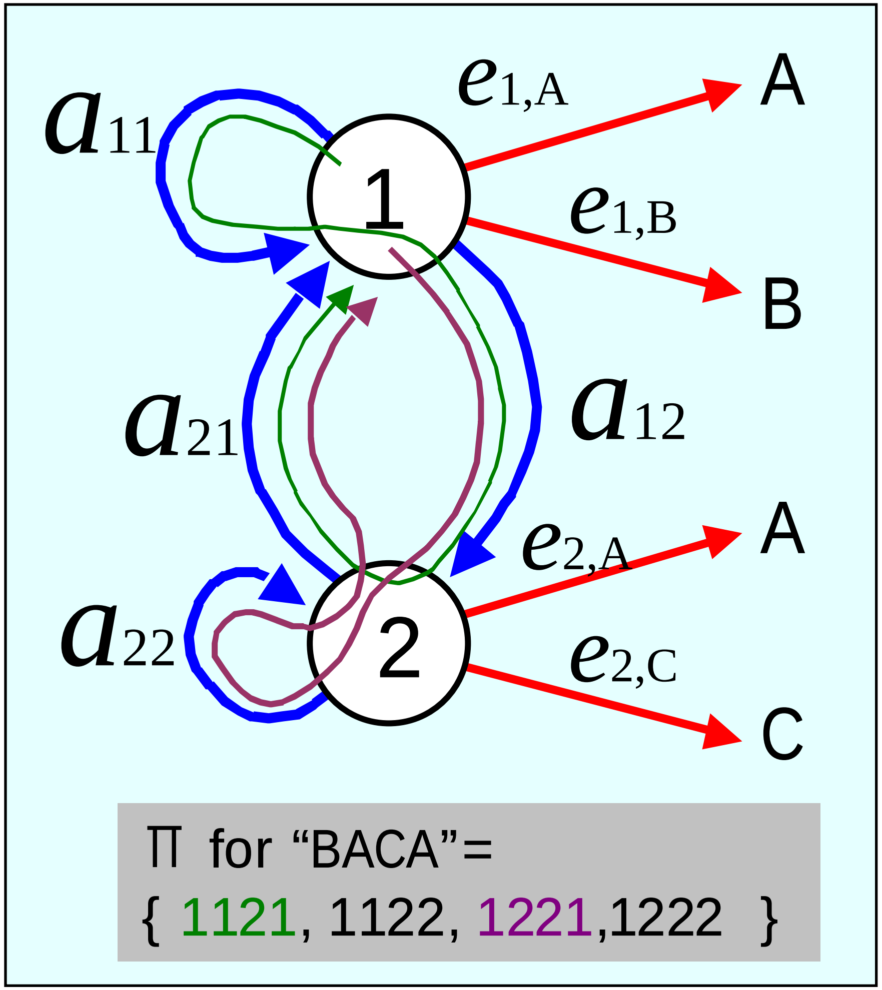
$$\Pr(x,\pi|\theta)=\prod_{i=1}^{n} a_{\pi_{i-1}\pi_i} e_{\pi_i, x_i}$$

- Viterbiアルゴリズム

$$\max_{\pi}\left\{ \Pr(x,\pi|\theta) \right\}$$

- Forwardアルゴリズム

$$\sum_{\pi}\left\{ \Pr(x,\pi|\theta) \right\}$$



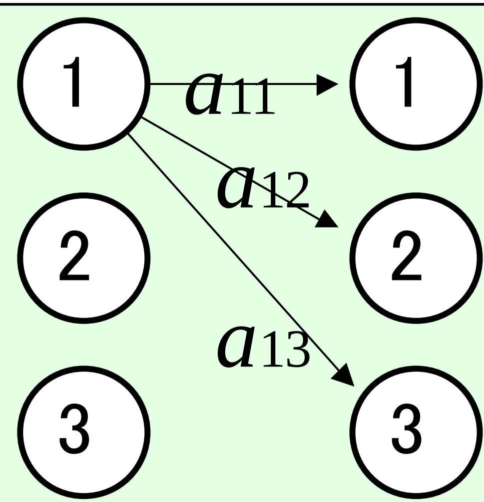
後向きアルゴリズム

- $b_k(i)=$
  $P(x_{i+1}\ldots x_L|\pi_i=k)$
  をDPにより計算
- $P(\pi_i=k|x)=$
  $f_k(i)b_k(i)/P(x)$

$$b_k(L)=a_{k0}$$

$$b_k(i)=\sum_k a_{kl}e_l(x_{i+1})b_k(i+1)$$

$$P(x)=\sum_k a_{0l}e_l(x_1)b_l(1)$$



$$b_1(i)=$$
$$(a_{11}e_1(x_{i+1})b_1(i+1)+$$
$$a_{12}e_2(x_{i+1})b_2(i+1)+$$
$$a_{13}e_3(x_{i+1})b_3(i+1))$$

HMMに対するEMアルゴリズム（Baum-Welchアルゴリズム）

A_{kl}: a_{kl}が使われる回数の期待値　　x^j : j番目の配列

$E_k(b)$: 文字bが状態kから現れる回数の期待値

$$A_{kl} = \sum_j \frac{1}{P(x^j)} \sum_i f_k^j(i) a_{kl} e_l(x_{i+1}^j) b_l^j(i+1)$$

$$E_k(b) = \sum_j \frac{1}{P(x^j)} \sum_{\{i \mid x_j^j = b\}} f_k^j(i) b_k^j(i)$$

パラメータの更新式

$$\hat{a}_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}} \qquad \hat{e}_k(b) = \frac{E_k(b)}{\sum_{b'} E_k(b')}$$

Baum-WelchのEMによる解釈

$$P(x,\pi\mid\theta)=\prod_{k=1}^{M}\prod_{b}\left[e_k(b)\right]^{E_k(b,\pi)}\prod_{k=0}^{M}\prod_{l=1}^{M}a_{kl}^{A_{kl}(\pi)}$$ および

$$Q(\theta\mid\theta^t)=\sum_{\pi}P(\pi\mid x,\theta^t)\log P(x,\pi\mid\theta)$$ より、

$$Q(\theta\mid\theta^t)=\sum_{\pi}P(\pi\mid x,\theta^t)\times\left[\sum_{k=1}^{M}\sum_{b}E_k(b,\pi)\log e_k(b)+\sum_{k=0}^{M}\sum_{l=1}^{M}A_{kl}(\pi)\log a_{kl}\right]$$

$$=\sum_{k=1}^{M}\sum_{b}E_k(b)\log e_k(b)+\sum_{k=0}^{M}\sum_{l=1}^{M}A_{kl}\log a_{kl}$$

ここで$\sum_i p_i\log q_i$は$q_i=p_i$の時、最大より、

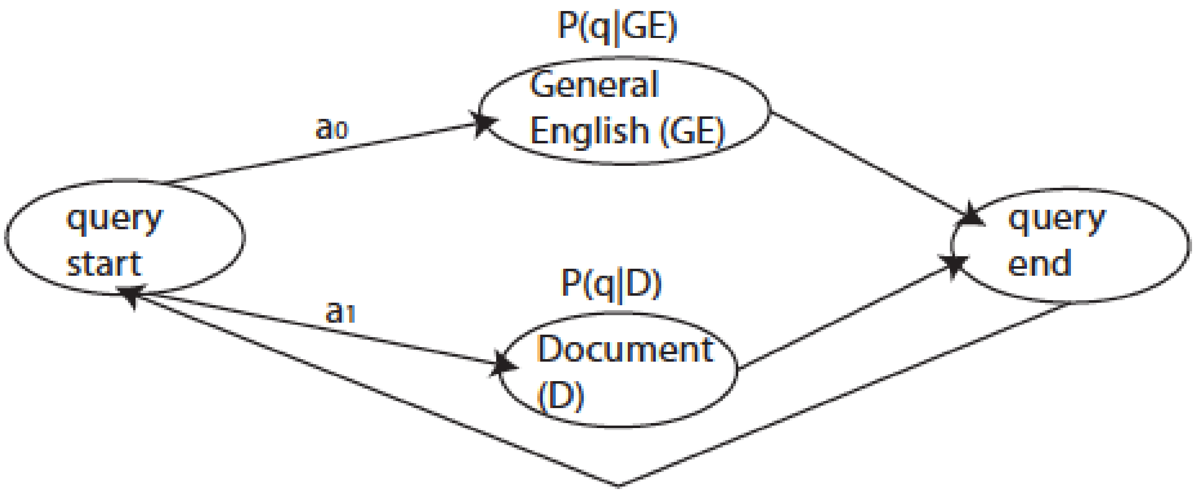
$$e_k(b)=\frac{E_k(b)}{\sum_{b'}E_k(b')},\quad a_{kl}=\frac{A_{kl}}{\sum_{l'}A_{kl'}}$$

情報検索

- インターネット上での文書（ドキュメント）は膨大になってきた
- 情報検索(Information Retrieval)
  - ユーザーが求める情報をこの膨大なドキュメント数から瞬時に検索する技術は大変重要である
- 検索エンジン
  - 多くの検索エンジンではユーザーが検索したい対象を複数のキーワードで入力できるようになっている
  - ユーザーが検索エンジンに入力する複数のキーワードをクエリー(query)という

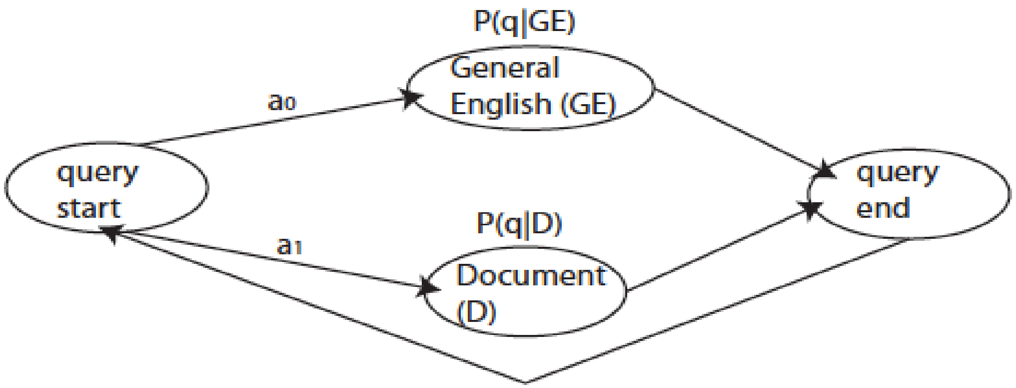
情報検索

- 情報検索のプロセス
- クエリーが観測シンボル、ドキュメントが隠れ状態とする隠れマルコフモデル(HMM)としてモデル化できる



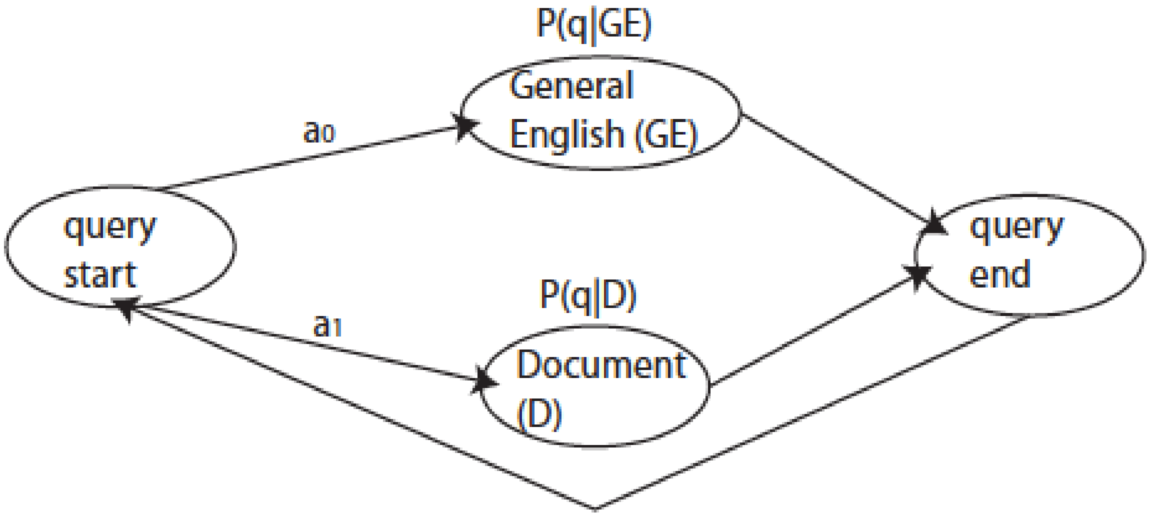
情報検索

- 4状態からなる隠れマルコフモデル
- クエリーに含まれているキーワードq
- ドキュメントDから生成される確率はP(q|D)
- qがどのドキュメントにも含まれそうな一般の英単語（例えば助詞）から生成される確率P(q|GE)
- 初期状態からGeneralEnglish状態への遷移確率a0
- 初期状態からドキュメント状態への遷移確率a1
- 初期状態をquery start
- 最終状態をquery end



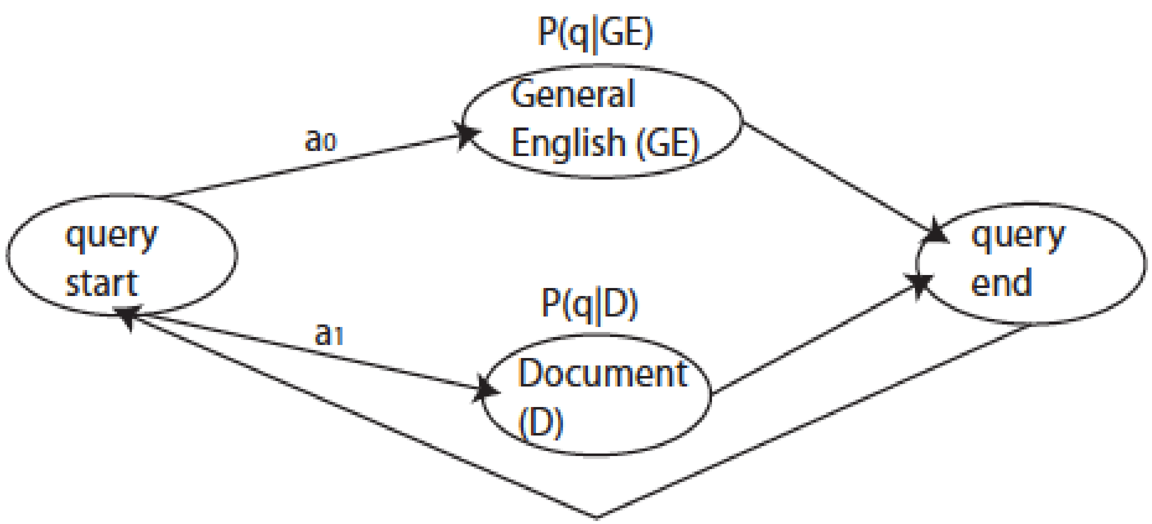
情報検索

- a0とa1が満たすべき制約は何か？



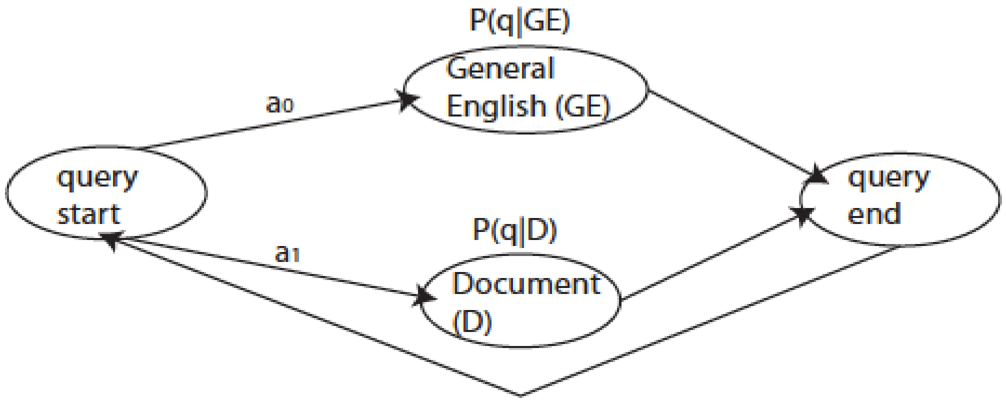
情報検索

- a0とa1が満たすべき制約は何か？



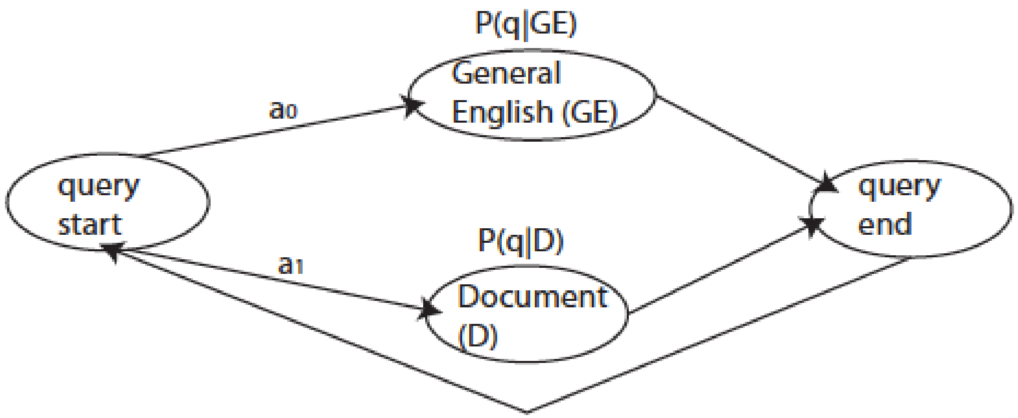
情報検索

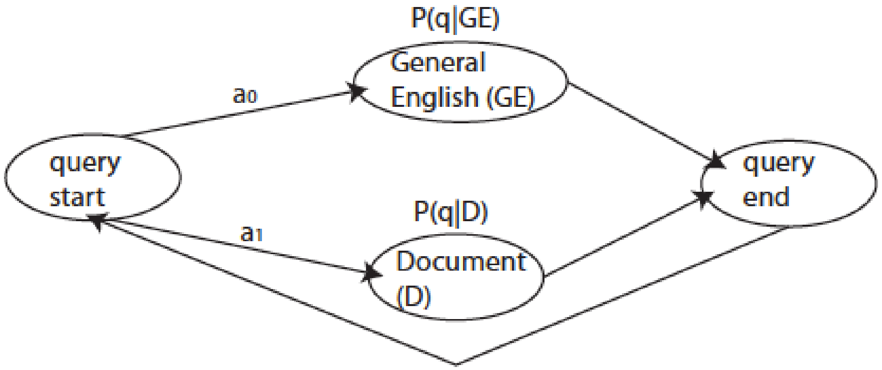
- 情報を検索する際にはクエリーqが与えられ、qに関連するドキュメントをその関連度(relevancy)でランクづけて表示する必要がある。
- そのために事後確率P(D|q)を用いることができる。
- ベイス定理を用い、P(D|q)をP(q|D), P(D)とP(q)で表してみよう。



情報検索

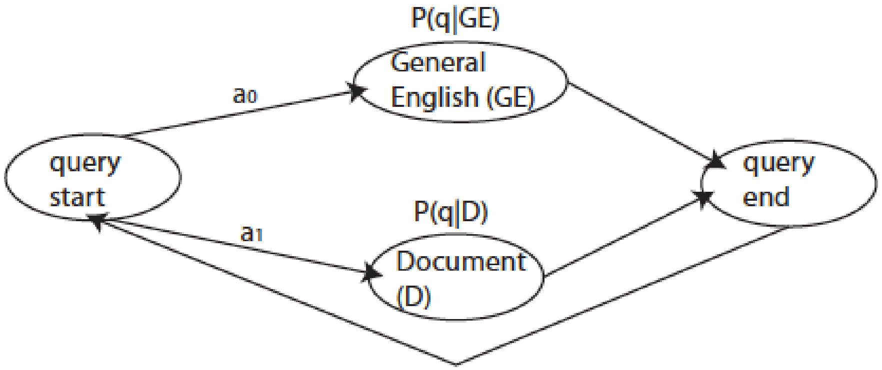
- 隠れマルコフモデルのパラメータであるa0、a1はそれぞれのドキュメントDに依存しているためドキュメント数が増えると内部パラメータ数も増えてしまうという欠点がある。
- パラメータ数がこのようにデータにつれて増えてしまうと学習の場合はどのような問題があるか？





レポート課題

- 図のマルコフモデルにおけるパラメータ推定を簡単化するために次の二つの仮定をおく。
  - 状態間の遷移確率はドキュメントに依存しないとする。これによってa_0,a_1は全てのドキュメントに関して同じものとなる。
  - パラメータ推定にEMアルゴリズムを用いず、最尤度推定を用いる。



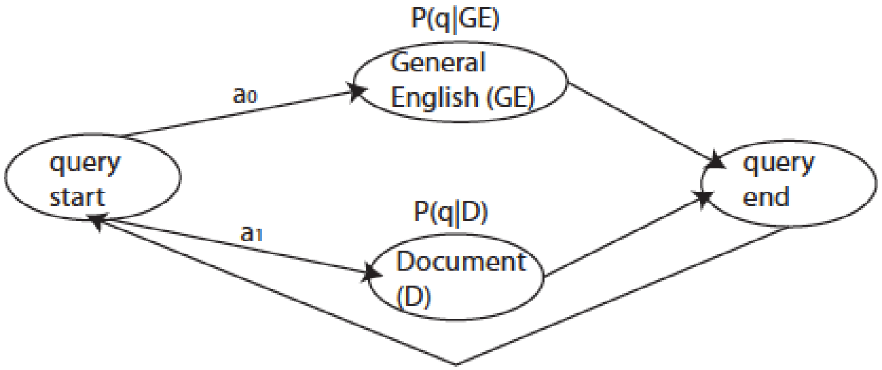
レポート課題

- 以上の仮定より、あるドキュメントD_kとクエリーqについて$P(q|D_k)$と$P(q|GE)$を次のように計算できる。

$$P(q|D_k) = \frac{D_k\text{の中の } q \text{ の出現回数}}{D_k\text{の中の全単語数}}$$

$$P(q|GE) = \frac{\sum_k D_k\text{の中の } q \text{ の出現回数}}{\sum_k D_k\text{の中の全単語数}}$$

レポート課題



1. 複数キーワード q からなるクエリーQ が与えられた場合、あるドキュメント D_kがそのクエリーQ に関連する確率 $P(Q|D_k)$を次の式で与えられることを示せ。但し、Q の中のキーワード q は全て独立だと仮定する。

$$P(Q|D_k) = \prod_{q \in Q} (a_0 P(q|GE) + a_1 P(q|D_k))$$

2. 下記のサンプルプログラムは図１のマルコフモデルを使った全文検索エンジンを実装している。このプログラムはパラメータ a_1 とクエリーQ を入力とし、P(Q|D)を使って関連するドキュメントをランクつけて表示する。様々なクエリーを用い、パラメータ a_1 を変更した場合検索結果がどう変わるか実験せよ。（プログラムの詳細については解凍したときに現れる README.pdf を参照すること。）
http://www.iba.t.u-tokyo.ac.jp/~danushka/data/HMMSearchEngine.tgz

3. EM アルゴリズムを用い、パラメータ a_1 を推定するためのプログラムを書け。上記の実験で求めたパラメータの値と EM アルゴリズムで求めた値を比較せよ。