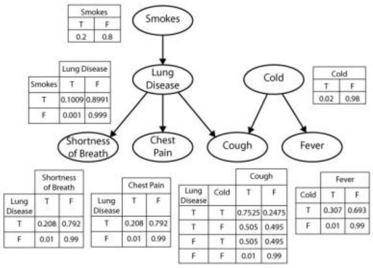
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

Hidden Markov Model

Department of Computer Science and Engineering
Indian Institute of Technology, Roorkee, India

Bayesian Network contd.



Smokes	
T	F
0.2	0.8

	Lung Disease	
Smokes	T	F
T	0.1009	0.8991
F	0.001	0.999

Cold	
T	F
0.02	0.98

	Shortness of Breath	
Lung Disease	T	F
T	0.208	0.792
F	0.01	0.99

	Chest Pain	
Lung Disease	T	F
T	0.208	0.792
F	0.01	0.99

		Cough	
Lung Disease	Cold	T	F
T	T	0.7525	0.2475
T	F	0.505	0.495
F	T	0.505	0.495
F	F	0.01	0.99

	Fever	
Cold	T	F
T	0.307	0.693
F	0.01	0.99

Markov Model

- Serves as the baseline model for sequential data.
- A random process of transitions from one state to another in a state space.
- It must posses Markov property i.e. the probability of the next state depends on current state not on previous states.

$$P(X_{t+1}=\mathbf{j} \mid X_0=k_0, X_1=k_1, \ldots , X_{t-1}=k_{t-1}, X_t=\mathbf{i}) = P(X_{t+1}=\mathbf{j} \mid X_t=\mathbf{i})$$

for all t = 0, 1, 2, … and for every possible state

Markov Models

- A discrete (finite) system:
  - N distinct states.
  - Begins (at time t=1) in some initial state(s).
  - At each time step (t=1,2,…) the system moves from current to next state (possibly the same as the current state) according to transition probabilities associated with current state.
- This kind of system is called a finite, or discrete Markov model

- **After Andrei Andreyevich Markov (1856 -1922)**

Discrete Markov Model: Example

- Discrete Markov Model with 5 states.
- Each a_{ij} represents the probability of moving from state i to state j
- The a_{ij} are given in a matrix A = {a_{ij}}
- The probability to start in a given state i is π_i , The vector π represents these start probabilities.

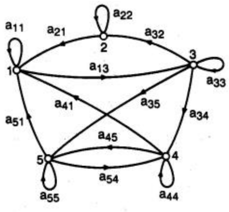
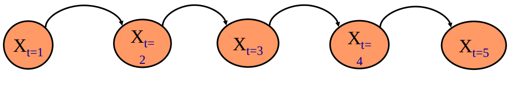


Figure 6.1 A Markov chain with five states (labeled 1 to 5) with selected state transitions.

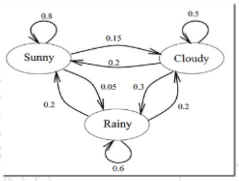
Markov Property

- **Markov Property: The state of the system at time t+1 depends only on the state of the system at time t**

$$P[X_{t+1} = x_{t+1} \mid X_t = x_t , X_{t-1} = x_{t-1} , \ldots , X_1 = x_1 , X_0 = x_0] \\ = P[X_{t+1} = x_{t+1} \mid X_t = x_t]$$



Markov Model Example



Today is sunny what's the prob. that tomorrow is sunny and the day after is rainy.

Markov Model Example

- P(w2=sunny,w3=rainy| w1=sunny)
  = P(w2=sunny|w1=sunny) * P(w3=rainy|w2=sunny,w1=sunny)
  = P(w2=sunny|w1=sunny) * P(w3=rainy|w2=sunny)
  = 0.8 * 0.05 = 0.04

2. P(w2=cloudy,w3=rainy| w1=sunny) ??

Discrete Markov Model - Example

- States – Rainy:1, Cloudy:2, Sunny:3
- Matrix A – $A = \{a_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}.$
- Problem – given that the weather on day 1 (t=1) is sunny(3), what is the probability for the observation O:

O	=	(sunny,	sunny,	sunny,	rain,	rain,	sunny,	cloudy,	sunny)
	=	(3,	3,	3,	1,	1,	3,	2,	3)
day		1	2	3	4	5	6	7	8

Discrete Markov Model – Example (cont.)

- The answer is -

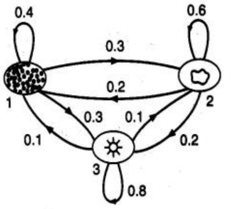


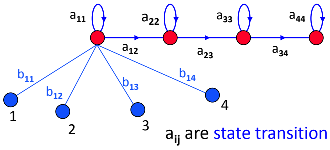
Figure 6.2 Markov model of the weather.

$$\begin{aligned} P(\mathrm{O}|\mathrm{Model}) &= P[3,3,3,1,1,3,2,3|\mathrm{Model}] \\ &= P[3]P[3|3]^2 P[1|3]P[1|1] \\ &\quad P[3|1]P[2|3]P[3|2] \\ &= \pi_3 \cdot (a_{33})^2 a_{31} a_{11} a_{13} a_{32} a_{23} \\ &= (1.0)(0.8)^2(0.1)(0.4)(0.3)(0.1)(0.2) \\ &= 1.536 \times 10^{-4} \end{aligned}$$

Hidden Markov Models

Often we face scenarios where states cannot be directly observed.

We need an extension: Hidden Markov Models



Observed phenomenon

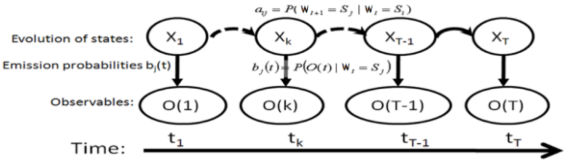
a_{ij} are state transition probabilities.

b_{ik} are observation (output) probabilities.

$b_{11} + b_{12} + b_{13} + b_{14} = 1$,
$b_{21} + b_{22} + b_{23} + b_{24} = 1$, etc.

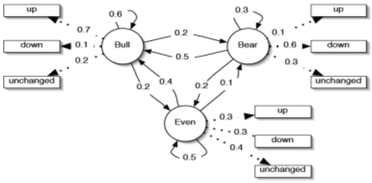
HMM parameter

- Transition Probability (a_{ij} =P($w_{t+1}|w_t$))
- Emission Probability (b_{jk}= P(V_t = k|w_t = j))
- Hidden States (W) and Visible States (V)

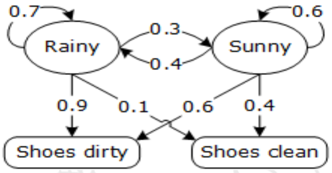

Transitional probabilities
$a_{ij} = P(W_{t+1} = S_j \mid W_t = S_i)$
Evolution of states:
X_1
X_k
X_{T-1}
X_T
Emission probabilities $b_j(t)$
$b_j(t) = P(O(t) \mid W_t = S_j)$
Observables:
O(1)
O(k)
O(T-1)
O(T)
Time:
t_1
t_k
t_{T-1}
t_T

HMM Example

- For observation sequence (up-down-down).
- We cannot say exactly what state sequence produces these observations thus state sequence is hidden.

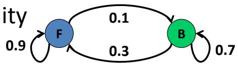


Example HMM



Markov Chain Example

- States: fair coin *F*, unfair (biased) coin *B*
- Discrete times: flip 1, 2, 3, …
- Initial probability: $\pi_F = 0.6$, $\pi_B = 0.4$
- Transition probability

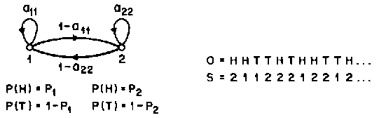


- Prob(FFBBFFFB)

$$P(O = \{FFBBFFFB\} \mid A, p)$$
$$= p_F \times a_{FF} \times a_{FB} \times a_{BB} \times a_{BF} \times a_{FF} \times a_{FF} \times a_{FB}$$
$$= 0.6´\ 0.9´\ 0.1´\ 0.7´\ 0.3´\ 0.9´\ 0.9´\ 0.1 = 9.19´\ 10^{-4}$$

Hidden Markov Model

- Coin toss example



- Coin transition is a Markov chain
- Probability of H/T depends on the coin used
- Observation of H/T is a hidden Markov chain (coin state is hidden)

Hidden Markov Model

- Elements of an HMM (coin toss)
  - N, the number of states (F / B)
  - M, the number of distinct observation (H / T)

$\lambda \left\{ \begin{array}{l} \text{– A = \{}a_{ij}\text{\} state transition probability} \quad A = \begin{bmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{bmatrix} \\ \text{– B = \{}b_j(k)\text{\} emission probability} \\ \quad b_F(H) = 0.5, b_F(T) = 0.5 \\ \quad b_B(H) = 0.8, b_B(T) = 0.2 \\ \text{– } \pi = \{\pi_i\} \text{ initial state distribution} \end{array} \right.$

  - $\pi_F = 0.4$, $\pi_B = 0.6$

HMM Applications

- Stock market: bull/bear market hidden Markov chain, stock daily up/down observed, depends on big market trend
- Speech recognition: sentences & words hidden Markov chain, spoken sound observed (heard), depends on the words
- Bioinformatics: sequence motif finding, gene prediction, genome copy number change, protein structure prediction, protein-DNA interaction prediction

Basic Problems for HMM

1. Given λ, how to compute P(O|λ) observing sequence O = $O_1O_2…O_T$
   - Probability of observing HTTHHHT …
   - Forward procedure, backward procedure
2. Given observation sequence O = $O_1O_2…O_T$ and λ, how to choose state sequence Q = $q_1q_2…q_t$
   - What is the hidden coin behind each flip
   - Forward-backward, Viterbi
3. How to estimate λ =(A,B,π) so as to maximize P(O|λ)
   - How to estimate coin parameters λ
   - Baum-Welch (Expectation maximization)

Problem 1: P(O|λ)

- Suppose we know the state sequence Q

$$P(O\,|\,Q,\lambda)=b_{q1}(O_1)b_{q2}(O_2)...b_{qT}(O_T)$$

  - O = H T T H H H T
  - Q = F F B F F B B

$$P(O\,|\,Q,\lambda)=b_F(H)b_F(T)b_B(T)b_F(H)b_F(H)b_B(H)b_B(T)$$
$$=0.5 \quad \times 0.5 \ \times 0.2 \quad \times 0.5 \quad \times 0.5 \ \times 0.8 \quad \times 0.2$$

  - Q = B F B F B B B

$$P(O\,|\,Q,\lambda)=b_B(H)b_F(T)b_B(T)b_F(H)b_B(H)b_B(H)b_B(T)$$
$$=0.8 \quad \times 0.5 \ \times 0.2 \quad \times 0.5 \quad \times 0.8 \ \times 0.8 \quad \times 0.2$$

- Each given path Q has a probability for O

Problem 1: P(O|λ)

- What is the prob of this path Q?

$$P(Q\,|\,\lambda)$$

  - Q = F F B F F B B

$$P(Q\,|\,\lambda)=\pi_F\ a_{FF}\ a_{FB}\ a_{BF}\ a_{FF}\ a_{FB}\ a_{BB}$$
$$=0.6\times 0.9\times 0.1\times 0.3\times 0.9\times 0.1\times 0.7$$

  - Q = B F B F B B B

$$P(Q\,|\,\lambda)=\pi_B\ a_{BF}\ a_{FB}\ a_{BF}\ a_{FB}\ a_{BB}\ a_{BB}$$
$$=0.4\times 0.3\times 0.1\times 0.3\times 0.1\times 0.7\times 0.7$$

- Each given path Q has its own probability

Problem 1: P(O|λ)

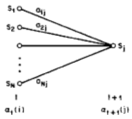
- Therefore, total pb of O = HTTHHHT
- Sum over all possible paths Q: each Q with its own pb multiplied by the pb of O given Q

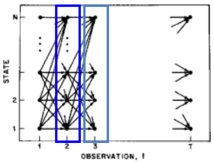
$$P(O\,|\,\lambda)=\sum_{all\ Q}P(O,Q\,|\,\lambda)=\sum_{all\ Q}P(Q\,|\,\lambda)P(O\,|\,Q,\lambda)$$

- For path of N long and T hidden states, there are T^N paths, unfeasible calculation

Solution to Prob1:Forward Procedure

- Use dynamic programming
- Summing at every time point
- Keep previous subproblem solution to speed up current calculation



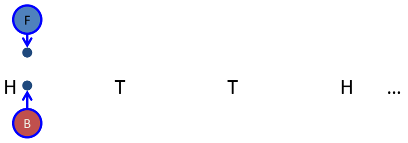


Forward Procedure

- Coin toss, O = HTTHHHT
- Initialization $\alpha_1(i)=\pi_i b_i(O_1)$

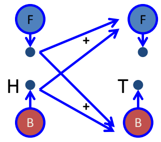
$$a_1(F)=p_F b_F(H)=0.4´\ \ 0.5=0.2$$
$$a_1(B)=p_B b_B(H)=0.6´\ \ 0.8=0.48$$

  - Pb of seeing H_1 from F_1 or B_1



Forward Procedure

- Coin toss, O = HTTHHHT
- Initialization $\alpha_1(i) = \pi_i b_i(O_1)$
- Induction $\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) a_{ij}] b_j(O_{t+1})$
  - Pb of seeing T_2 from F_2 or B_2



F_2 could come from F_1 or B_1

Each has its pb, add them up

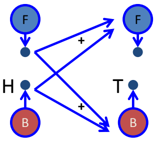
T H …

Forward Procedure

- Coin toss, O = HTTHHHT
- Initialization $\alpha_1(i) = \pi_i b_i(O_1)$
- Induction $\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) a_{ij}] b_j(O_{t+1})$

$$\alpha_2(F) = (\alpha_1(F) a_{FF} + \alpha_1(B) a_{BF}) b_F(T) = (0.2 \times 0.9 + 0.48 \times 0.3) \times 0.5 = 0.162$$

$$\alpha_2(B) = (\alpha_1(F) a_{FB} + \alpha_1(B) a_{BB}) b_B(T) = (0.2 \times 0.1 + 0.48 \times 0.7) \times 0.2 = 0.0712$$



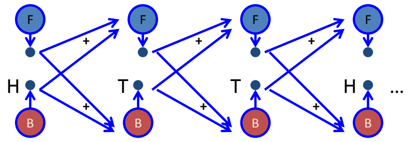
T H …

Forward Procedure

- Coin toss, O = HTTHHHT
- Initialization $\alpha_1(i) = \pi_i b_i(O_1)$
- Induction $\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) a_{ij}] b_j(O_{t+1})$

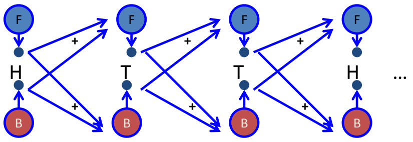
$$\alpha_3(F) = [0.162 \times 0.9 + 0.0712 \times 0.3] \times 0.5 = 0.08358$$

$$\alpha_3(B) = [0.162 \times 0.1 + 0.0712 \times 0.7] \times 0.2 = 0.013208$$



Forward Procedure

- Coin toss, O = HTTHHHT
- Initialization $\alpha_1(i) = \pi_i b_i(O_1)$
- Induction $\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) a_{ij}] b_j(O_{t+1})$
- Termination $P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_t(i) \quad = \alpha_4(F) + \alpha_4(B)$

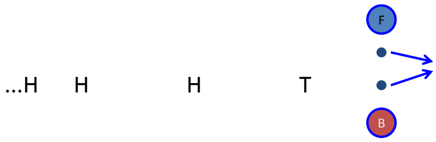


Solution to Prob1: Backward Procedure

- Coin toss, O = HTTHHHT
- Initialization $\beta_{T*}(i) = 1$

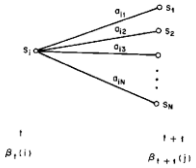
$$\beta_{T*}(F) = \beta_{T*}(B) = 1$$

- Pb of coin to see certain flip after it



Backward Procedure

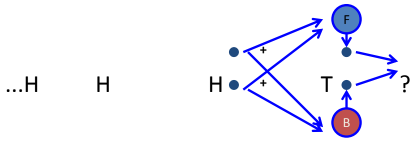
- Coin toss, O = HTTHHHT
- Initialization $\beta_{T*}(i) = 1$
- Induction $\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$
- Pb of coin to see certain flip after it



Backward Procedure

- Coin toss, O = HTTHHHT
- Initialization $\beta_{T*}(i) = 1$
- Induction $\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$

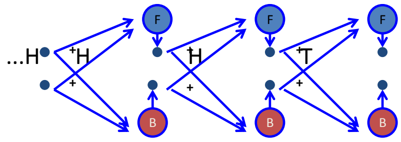
$$\beta_{T-1}(F) = a_{FF} b_F(T) \times 1 + a_{FB} b_B(T) \times 1 = 0.9 \times 0.5 + 0.1 \times 0.2 = 0.47$$
$$\beta_{T-1}(B) = a_{BF} b_F(T) \times 1 + a_{BB} b_B(T) \times 1 = 0.3 \times 0.5 + 0.7 \times 0.2 = 0.29$$



Backward Procedure

- Coin toss, O = HTTHHHT
- Initialization $\beta_{T*}(i) = 1$
- Induction $\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$

$$\beta_{T-2}(F) = a_{FF} b_F(H) \times \beta_{T-1}(F) + a_{FB} b_B(H) \times \beta_{T-1}(B) = 0.9 \times 0.5 \times 0.47 + 0.1 \times 0.8 \times 0.29 = 0.2347$$
$$\beta_{T-2}(B) = a_{BF} b_F(H) \times \beta_{T-1}(F) + a_{BB} b_B(H) \times \beta_{T-1}(B) = 0.3 \times 0.5 \times 0.47 + 0.7 \times 0.8 \times 0.29 = 0.2329$$



Backward Procedure

- Coin toss, O = HTTHHHT
- Initialization $\beta_{T*}(i) = 1$
- Induction $\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$
- Termination $\beta_0(*) = \pi_F b_F(H) \beta_1(F) + \pi_B b_B(H) \beta_1(B)$
- Both forward and backward could be used to solve problem 1, which should give identical results

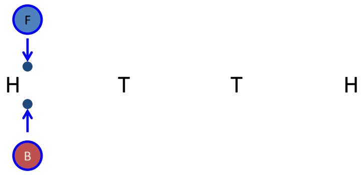
Solution to Problem 2: Viterbi Algorithm

Report the path that is most likely to give the observations

- Initiation $\delta_1(i) = \pi_i b_i(O_1)$
  $\psi_1(i) = 0$
- Recursion $\delta_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}] b_j(O_t)$
  $\psi_t(j) = \arg\max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}]$
- Termination $P^* = \max_{1 \le i \le N} [\delta_T(i)]$
  $q_T^* = \arg\max_{1 \le i \le N} [\delta_T(i)]$
- Path (state sequence) backtracking

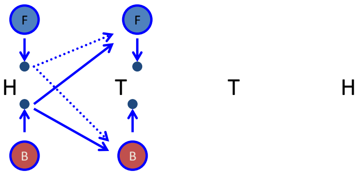
$$q_t^* = \psi_{t+1}(q_{t+1}^*), t = T-1, T-2, \ldots, 1$$

Viterbi Algorithm



Viterbi Algorithm

- Max instead of +, keep track of path
- Best path (instead of all path) up to here



Viterbi Algorithm

- Observe: HTTHHHT
- Initiation $\delta_1(i) = \pi_i b_i(O_1)$
  $\psi_1(i) = 0$

$$\delta_1(F) = 0.4 \times 0.5 = 0.2$$
$$\delta_1(B) = 0.6 \times 0.8 = 0.48$$

Viterbi Algorithm

- Observe: HTTHHHT
- Initiation $\delta_1(i) = \pi_i b_i(O_1)$
  $\psi_1(i) = 0$
- Recursion $\delta_t(j) = \max_{1 \le i \le N}[\delta_{t-1}(i) a_{ij}] b_j(O_t)$
  $\psi_t(j) = \arg\max_{1 \le i \le N}[\delta_{t-1}(i) a_{ij}]$

Max instead of +, keep track path

$$\delta_2(F) = [\max_i(\delta_1(F) a_{FF}, \delta_1(B) a_{BF})] b_F(T) = \max(\boxed{0.18,}0.144) \times 0.5 = 0.09$$
$$\delta_2(B) = [\max_i(\delta_1(F) a_{FB}, \delta_1(B) a_{BB})] b_B(T) = \max(0.02,\boxed{0.336)} \times 0.2 = 0.0672$$
$$\varphi_2(F) = F, \varphi_2(B) = B$$

Viterbi Algorithm

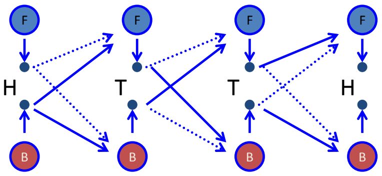
- Observe: HTTHHHT
- Initiation $\delta_1(i) = \pi_i b_i(O_1)$
  $\psi_1(i) = 0$
- Recursion $\delta_t(j) = \max_{1 \le i \le N}[\delta_{t-1}(i) a_{ij}] b_j(O_t)$
  $\psi_t(j) = \arg\max_{1 \le i \le N}[\delta_{t-1}(i) a_{ij}]$

Max instead of +, keep track path

$$\delta_3(F) = [\max_i(\delta_2(F) a_{FF}, \delta_2(B) a_{BF})] b_F(T)$$
$$= \max(\boxed{0.09 \times 0.9},0.0672 \times 0.3) \times 0.5 = 0.0405$$
$$\delta_3(B) = [\max_i(\delta_2(F) a_{FB}, \delta_2(B) a_{BB})] b_B(T)$$
$$= \max(0.09 \times 0.1,\boxed{0.0672 \times 0.7)} \times 0.2 = 0.0094$$
$$\varphi_3(F) = F, \varphi_3(B) = B$$

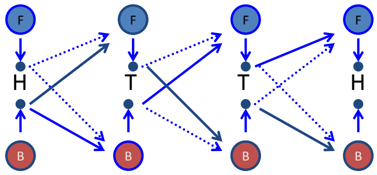
Viterbi Algorithm

- Max instead of +, keep track of path
- Best path (instead of all path) up to here



Viterbi Algorithm

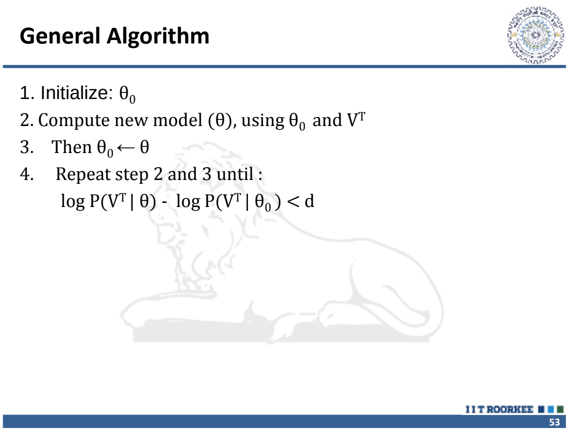
- Terminate, pick state that gives final best δ score, and backtrack to get path



- BFBB most likely to give HTTH

Solution to Problem 3

- No optimal way to do this, so find local maximum
- Baum-Welch algorithm (equivalent to expectation-maximization)
  - Random initialize λ =(A,B,π)
  - Run Viterbi based on λ and O
  - Update λ =(A,B,π)
    - π: % of F vs B on Viterbi path
    - A: frequency of F/B transition on Viterbi path
    - B: frequency of H/T emitted by F/B



General Algorithm

1. Initialize: θ_0
2. Compute new model (θ), using θ_0 and V^T
3. Then $\theta_0 \leftarrow \theta$
4. Repeat step 2 and 3 until :
   $\log P(V^T \mid \theta) - \log P(V^T \mid \theta_0) < d$