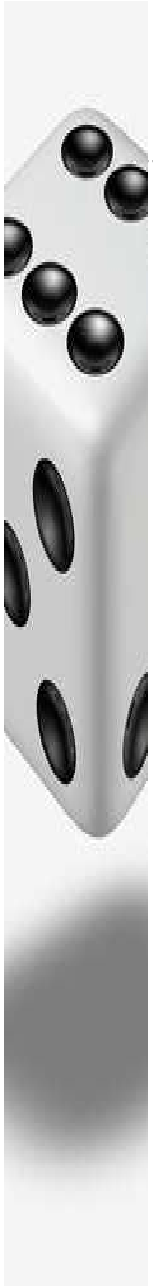
Hidden Markov Models

C SC 645
Raquel Torres
02.05.09



Outline

HMM
- Elements
- Examples

The three basic problems (and its solutions)
- Problem 1 – Forward-backward algorithm
- Problem 2 – Viterbi algorithm
- Problem 3 - Sum product / Baum-Welch

More Examples


corbis
What's next?

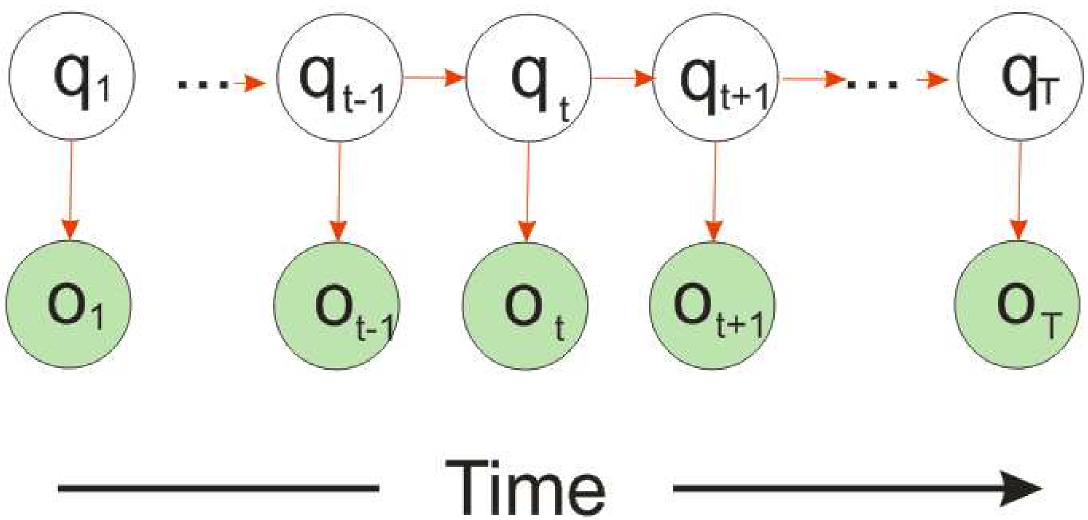
Hidden Markov Models

A hidden Markov model (HMM) is a discrete-sate model in which the system being modeled is assumed to be a Markov process with unknown parameters.

$S=\{S_1, S_2 \ldots S_N\}$ Individual states

q_t is the state at time t

$O= o_1, o_2 \ldots o_T$ Observation sequence

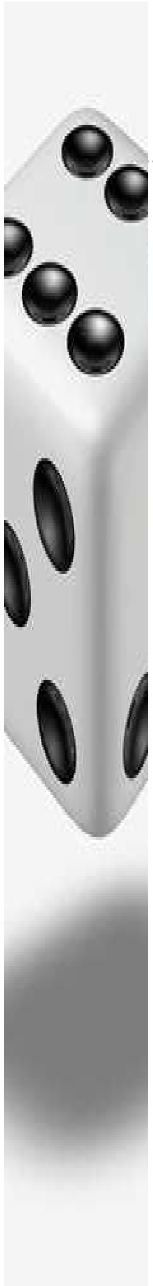


Hidden Markov Models

Examples of HMM:

- Text written by Shakespeare in some parts has been edited by a monkey
- A casino has two dice, one loaded and the other not. Toggles between them.

Case	Observations	Hidden state
Text	Alphabet	Shakespeare/monkey
Dice	1-6	Fair (F) / loaded (L)



Elements of a HMM

An HMM is completely defined by:

1) N, the number of states in the model $S=\{S_1, S_2 \ldots S_N\}$
2) M, the number of distinct observation symbols per state (An alphabet of symbols $V=\{v_1, v_2 \ldots v_M\}$)
3) The state transition probability distribution $A=(a_{ij})$ where

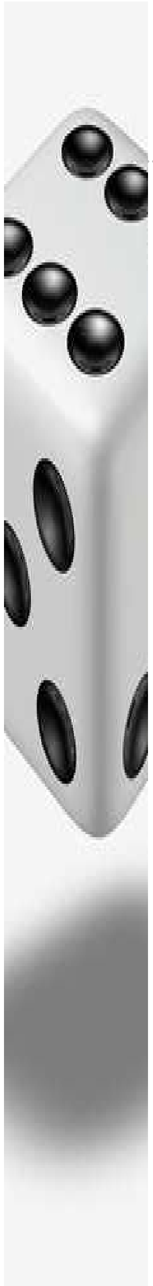
$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \qquad 1 \leq i, j \leq N.$$

4) Emission probability. The observation symbol probability distribution in state j, $B=\{b_J(k)\}$, where:

$$b_j(k) = P[v_k \text{ at } t | q_t = S_j], \qquad 1 \leq j \leq N \quad 1 \leq k \leq M.$$

5) The initial state distribution $\pi = \{\pi_i\}$ where:

$$\pi_i = P[q_1 = S_i], \qquad 1 \leq i \leq N.$$



Elements of a HMM

We will use the next notation to make reference to

The model:

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

Where

A= transition matrix.

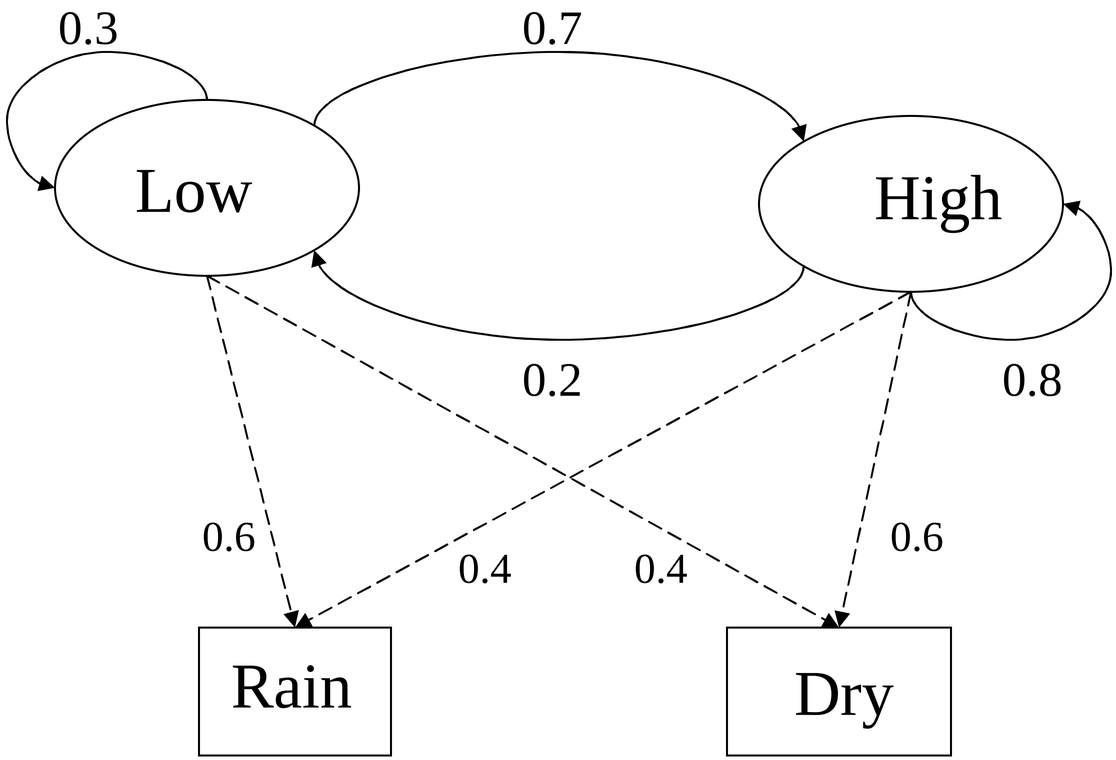
B= Emission matrix.

The probability of an observation sequence given a model.

$$P(O \mid \lambda)$$

Hidden Markov Models

Example 1



Transition probability matrix:

	Low	High
Low	0.3	0.7
High	0.2	0.8

Emission probability matrix:

	Rain	Dry
Low	0.6	0.4
High	0.4	0.3

Two states : ‘Low’ and ‘High’ atmospheric pressure.
Two observations : ‘Rain’ and ‘Dry’.
Initial probabilities: P(‘Low’)=0.4 , P(‘High’)=0.6 .

Hidden Markov Models

Example 1

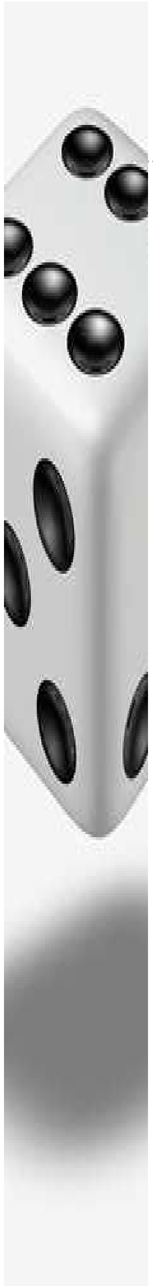
Suppose we want to calculate a probability of a sequence of observations in our example, {‘Dry’,’Rain’}.

Consider all possible hidden state sequences:

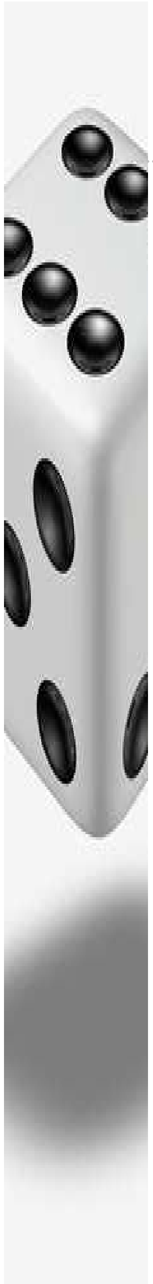
P({‘Dry’,’Rain’}) = P({‘Dry’,’Rain’} , {‘Low’,’Low’}) + P({‘Dry’,’Rain’} , {‘Low’,’High’}) + P({‘Dry’,’Rain’} , {‘High’,’Low’}) + P({‘Dry’,’Rain’} , {‘High’,’High’})

where first term is :

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



The Three
Basic Problems of HMM



The casino

A casino has two dice:

Fair die:

P (1) = P (2) = P (3) = P (5) = P (6) = 1 / 6

Loaded die:

P (1) = P (2) = P (3) = P (5) = 1 / 10

P (6) = 1 / 2

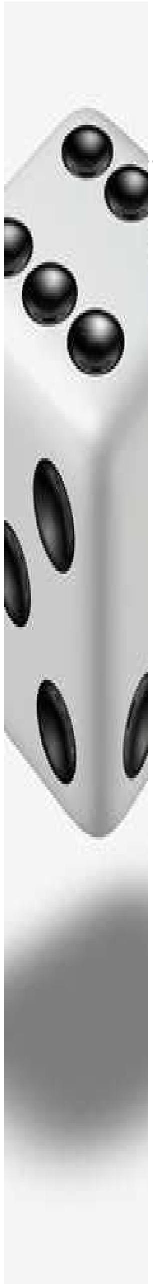
The casino alternates between the dice once every 20 turns

Game:

The player throws (a die always fair)

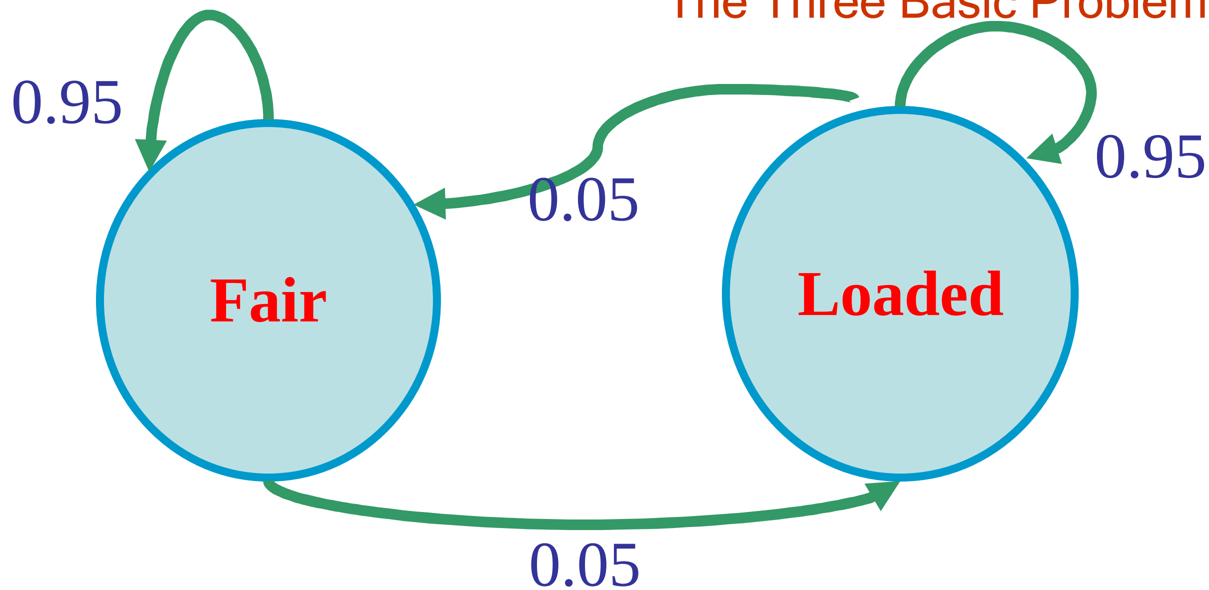
The casino throws (perhaps with the fair die, perhaps with a loaded)

The highest number wins



Hidden Markov Models

The Three Basic Problems of HMM



Emission probability

P(1|F) = 1/6
P(2|F) = 1/6
P(3|F) = 1/6
P(4|F) = 1/6
P(5|F) = 1/6
P(6|F) = 1/6

P(1|L) = 1/10
P(2|L) = 1/10
P(3|L) = 1/10
P(4|L) = 1/10
P(5|L) = 1/10
P(6|L) = 1/2

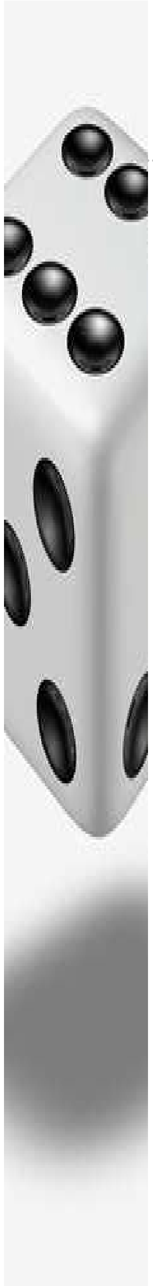
The Three Basic Problems of HMM

Given the next sequence:

124552646214614613613666166466163

6616366163616515615115146123562344

How likely is this sequence, given our model of how the casino works?



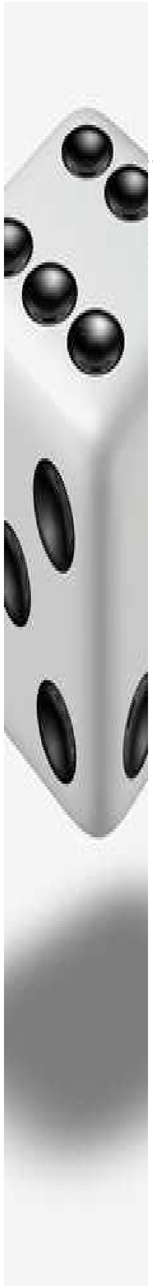
The Three Basic Problems of HMM

- Problem 1 (Evaluation): Given the observation sequence $O=o_1,\dots,o_T$ and an HMM model $\lambda=(A,B,\pi)$, how do we compute the probability of O given the model?

What is $P(O \mid \lambda)$?

The probability of an observation sequence O is the sum of the probabilities of all possible state sequences in the model.

Naïve computation is very expensive. Given T observations and N states, there are N^T possible state sequences.



The Three Basic Problems of HMM

Again, given the next sequence:

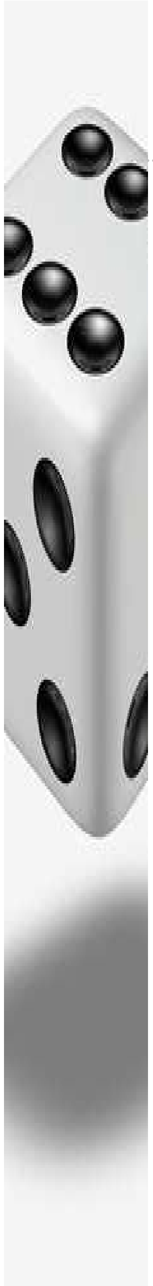
124552646214614613613666166466163

661636616361651561511514612356234

What portions of the sequence were generated by the loaded die and which by the fair die?

The Three Basic Problems of HMM

- Problem 2 (Decoding): Given the observation sequence $O=o_1,\ldots,o_T$ and an HMM model $\lambda = (A, B, \pi)$, how do we find the state sequence that best explains the observations?

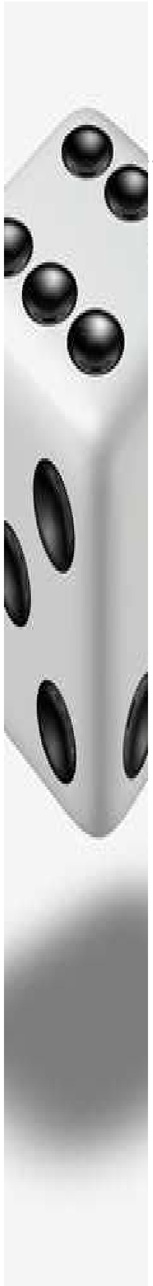


Again, given the next sequence:

124552646214614613613666166466163

661636616361651561511514612356234

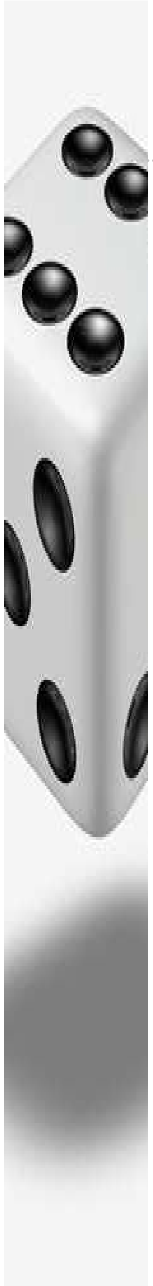
How the dice are loaded? How often the casino alternates between the dice?



Hidden Markov Models

The Three Basic Problems of HMM

- Problem 3 (Learning): How do we adjust the model parameters $\lambda = (A, B, \pi)$, to maximize $P(O \mid \lambda)$?



The forward-backward algorithm

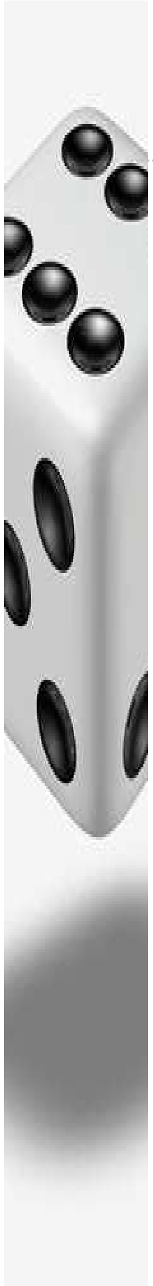
Problem 1: Evaluation Problem

Given a model and a sequence of observations, how do we compute the probability that the observed sequence was produced by the model?

The forward-backward algorithm

The algorithm comprises three steps:

1. computing forward probabilities
2. computing backward probabilities
3. computing smoothed values



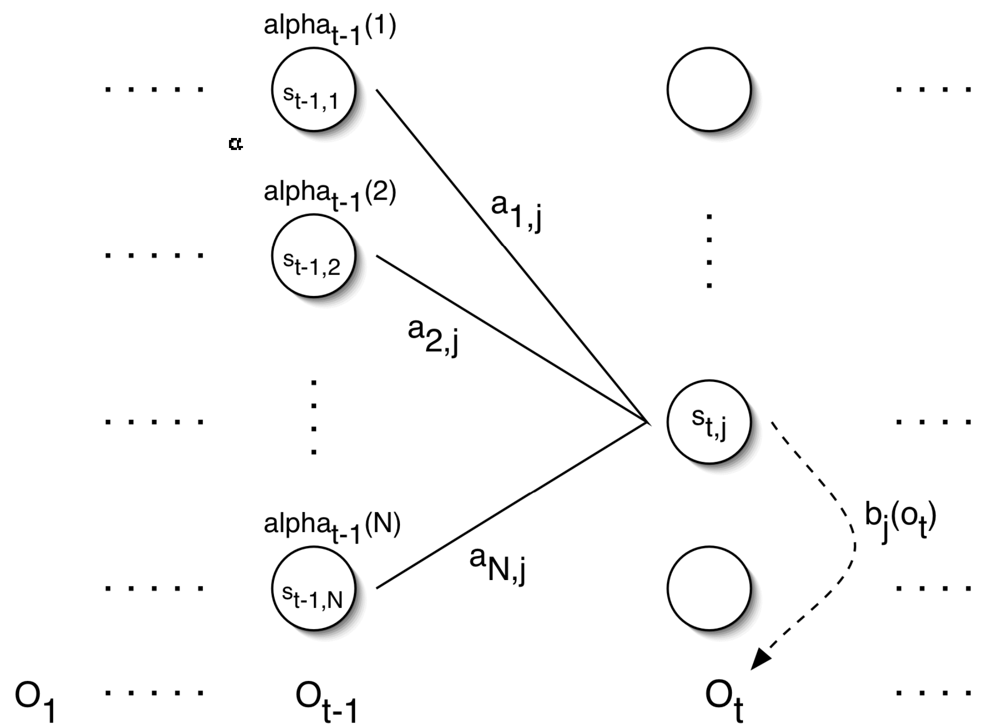
The forward probability

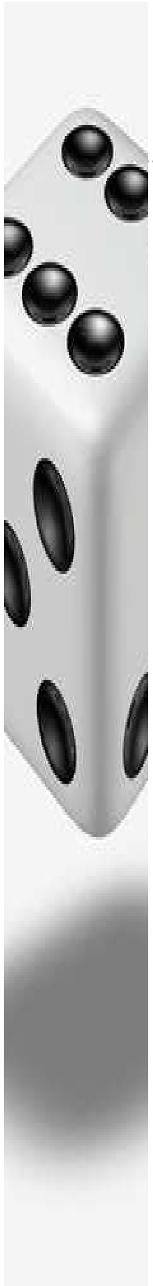
What is the probability that, given a model λ, at time t the state is i and the partial observation $o_1 \dots o_t$ has been generated?

$$\alpha_t(i) = P(o_1 \dots o_t, q_t = s_i \mid \lambda)$$

We reduce the complexity of calculating this probability by first calculating partial probabilities.

These represent the probability of getting to a particular state, s, at time t.





Hidden Markov Models

The forward-backward algorithm

- Initialization

$$\alpha_1(i) = \pi_i b_i(o_1) \quad 1 \le i \le N$$

- Induction

$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i)\, a_{ij} \right] b_j(o_t) \quad 2 \le t \le T, 1 \le j \le N$$

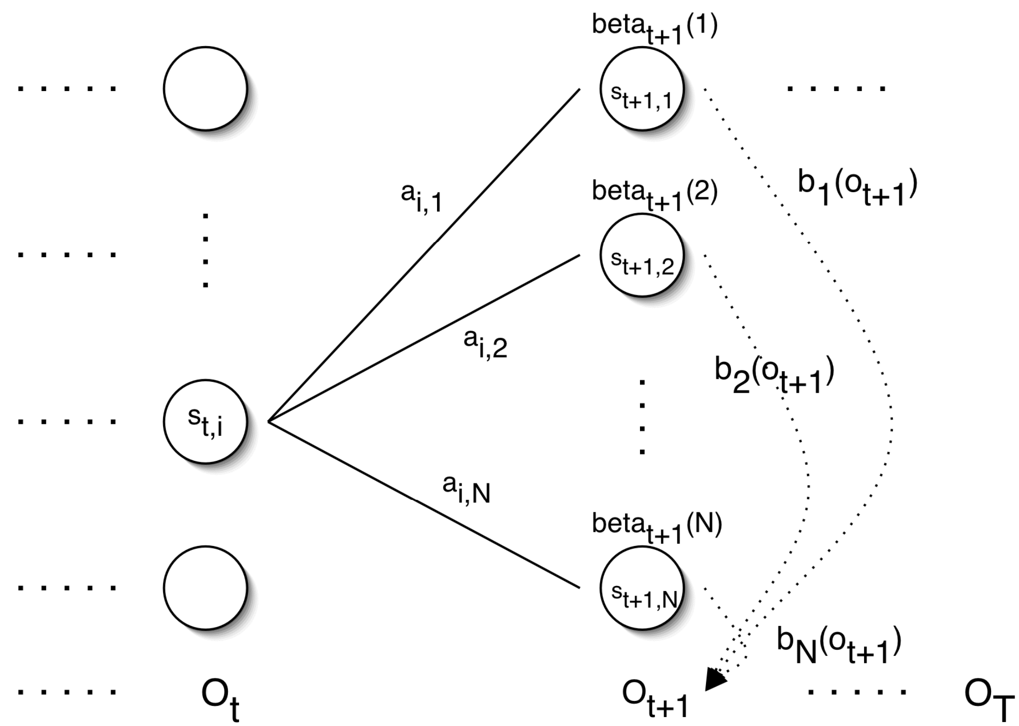
- Termination

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

The backward probability

The probability of the partial observation sequence from t+1 to the end, given state S_i at time t and the model λ .

$$\beta_t(i) = P(o_{t+1} \ldots o_T \mid q_t = s_i, \lambda)$$



The forward-backward algorithm

- Initialization

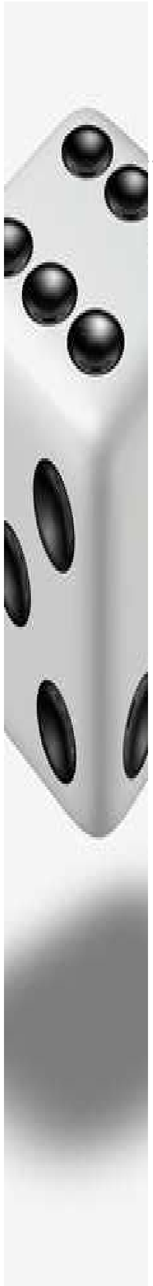
$$\beta_T(i) = 1, \quad 1 \le i \le N$$

- Induction

$$\beta_t(i) = \left[\sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \right] \; t = T-1 \ldots 1, 1 \le i \le N$$

- Termination

$$P(O \mid \lambda) = \sum_{i=1}^{N} \pi_i \beta_1(i)$$



- Problem 2 (Decoding): Finding the “optimal” state sequence associated with given observation sequence.

The Viterbi algorithm

- Similar to computing the forward probabilities, but instead of summing over transitions from incoming states, compute the maximum.

Forward:

$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i)\, a_{ij} \right] b_j(o_t)$$

Viterbi Recursion:

$$\delta_t(j) = \left[\max_{1 \le i \le N} \delta_{t-1}(i)\, a_{ij} \right] b_j(o_t)$$

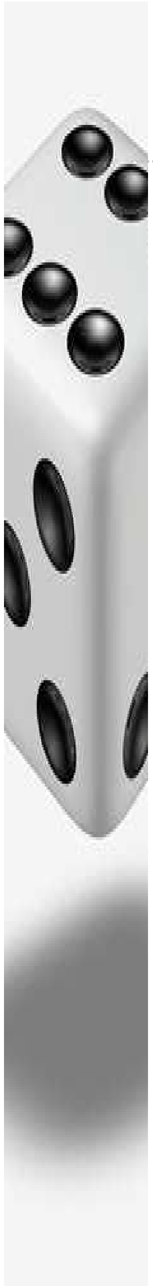
To implement the solution to problem 2, we define:

$$\gamma_t(i) = P(q_t = S_i | O, \lambda)$$

$$\gamma_t(i) = \frac{\alpha_t(i)\, \beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\, \beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\, \beta_t(i)}$$

We want to find the state sequence $Q=q_1 \ldots q_T$, such that

$$q_t = \underset{1 \le i \le N}{\operatorname{argmax}} [\gamma_t(i)],$$



The Viterbi algorithm

- Initialization:

$$\delta_1(i) = \pi_i b_j(o_1) \quad 1 \le i \le N$$

- Induction:

$$\delta_t(j) = \left[\max_{1\le i\le N} \delta_{t-1}(i)\, a_{ij}\right] b_j(o_t)$$

$$\psi_t(j) = \left[\arg\max_{1\le i\le N} \delta_{t-1}(i)\, a_{ij}\right] \quad 2 \le t \le T, 1 \le j \le N$$

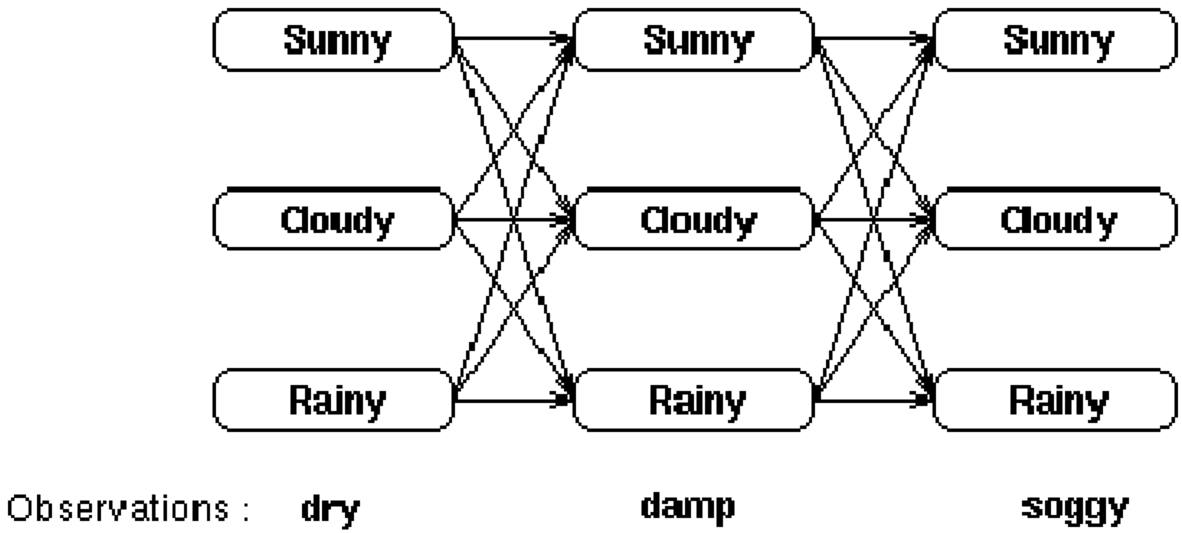
- Termination:

$$p^* = \max_{1\le i\le N} \delta_T(i) \qquad q_T^* = \arg\max_{1\le i\le N} \delta_T(i)$$

- Path (state sequence) backtracking:

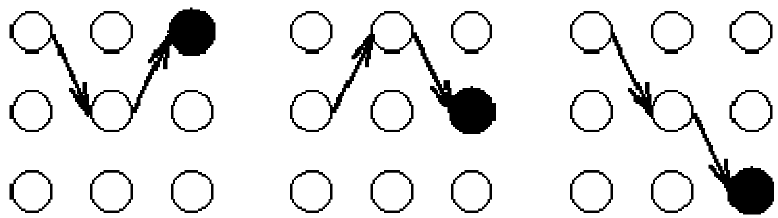
$$q_t^* = \psi_{t+1}(q_{t+1}^*) \quad t = T-1, \ldots, 1$$

The Viterbi algorithm



We will first define the partial probability , which is the probability of reaching a particular intermediate state in the trellis. We then show how these partial probabilities are calculated at t=1 and at t=n (> 1).

The Viterbi algorithm



These partial probabilities differ from those calculated in the forward algorithm since they represent the probability of the most probable path to a state at time t, and not a total.

In particular, each state at time t = T will have a partial probability and a partial best path. We find the overall best path by choosing the state with the maximum partial probability and choosing its partial best path

Example:
http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/viterbi_algorithm/s3_pg1.html

The Three Basic Problems of HMM

- Problem 3 (Learning): How do we adjust the model parameters $\lambda = (A, B, \pi)$, to maximize $P(O \mid \lambda)$?

Given an initial model λ, we can always find a model λ', such that

$$P(O \mid \lambda') \geq P(O \mid \lambda)$$

Hidden Markov Models

The sum-product / Baum-Welch algorithm

Use the forward-backward (or Baum-Welch) algorithm, which is a hill-climbing algorithm.

Using an initial parameter instantiation, the forward-backward algorithm iteratively re-estimates the parameters and improves the probability that given observation are generated by the new parameters.

The sum-product / Baum-Welch algorithm

Three parameters need to be re-estimated:

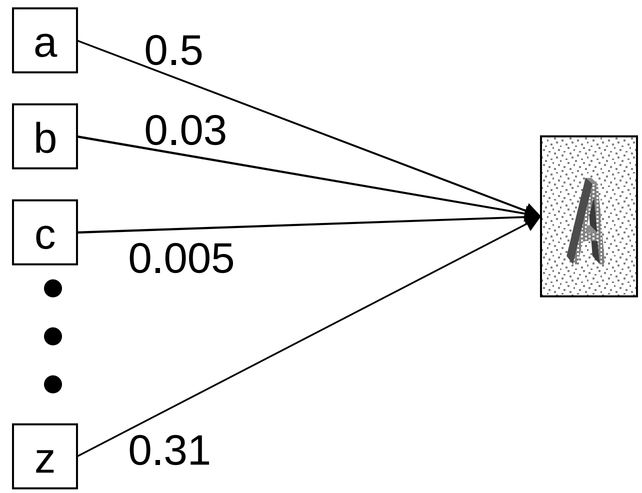
- Initial state distribution: π_i
    - Transition probabilities: $a_{i,j}$
    - Emission probabilities: $b_i(o_t)$

Example:
Word recognition (your turn)

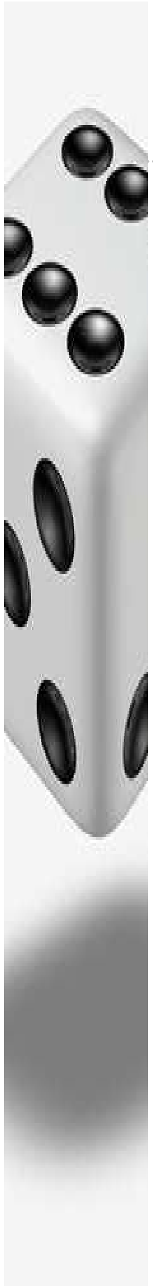
Example: Word recognition

Typed word recognition, assume all characters are separated.

Character recognizer outputs probability of the image being particular character, P(image|character).



Hidden state   Observation



Hidden Markov Models

Example: Word recognition

- Hidden states of HMM = characters.

- Observations = typed images of characters segmented from the image .
Note that there is an infinite number of observations

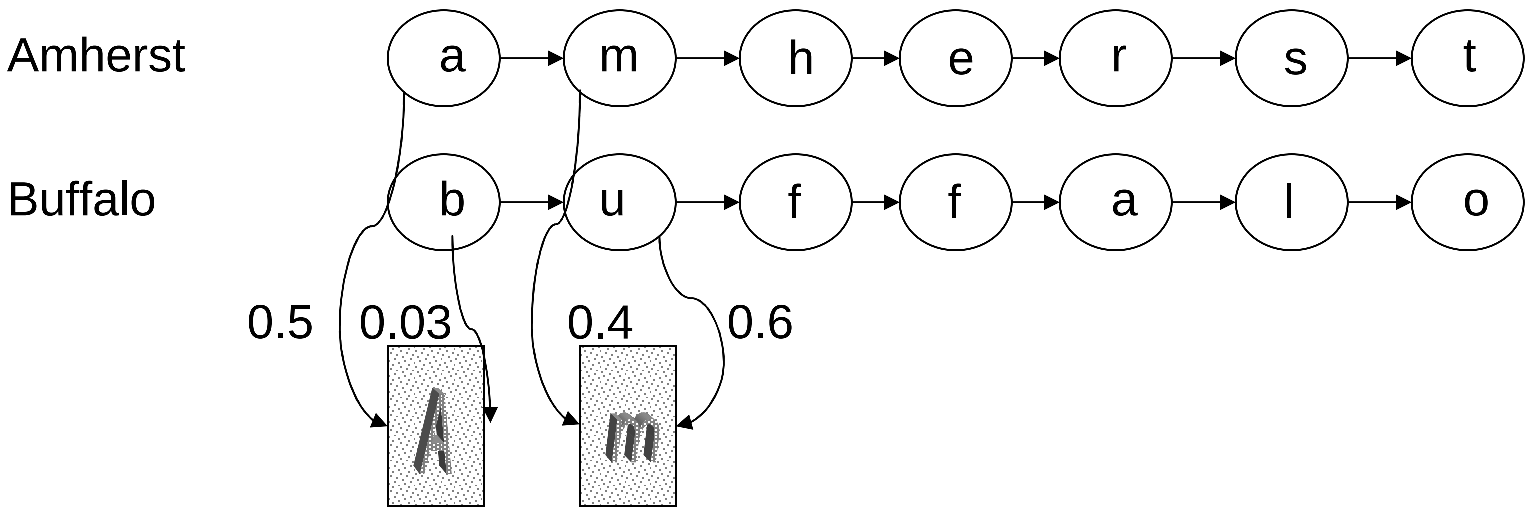
$$v_\alpha$$

- Emission probabilities = character recognizer scores.

$$B = \left(b_i(v_\alpha)\right) = \left(P(v_\alpha \mid s_i)\right)$$

Example: Word recognition

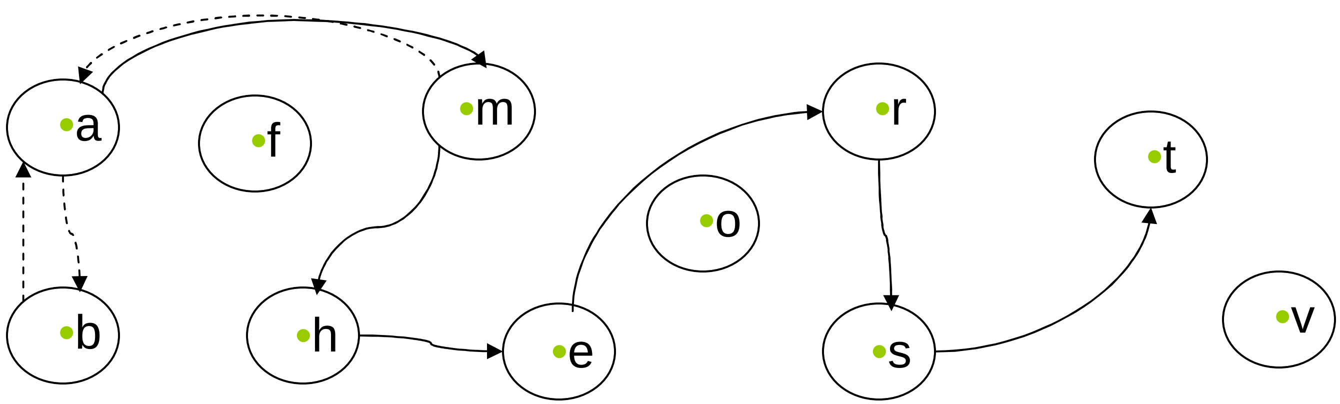
- If lexicon is given, we can construct separate HMM models for each lexicon word.



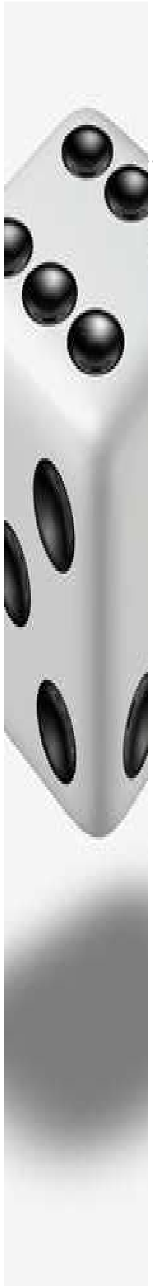
- Here recognition of word image is equivalent to the problem of evaluating few HMM models.
- This is an application of **Evaluation problem.**

Example: Word recognition

- We can construct a single HMM for all words.
- Hidden states = all characters in the alphabet.
- Transition probabilities and initial probabilities are calculated from language model.
- Observations and observation probabilities are as before.



- Here we have to determine the best sequence of hidden states, the one that most likely produced word image.
- This is an application of **Decoding problem.**



Discussion