CS 4375 – Introduction to Machine Learning

Hidden Markov Models

Erick Parolin

THE UNIVERSITY OF TEXAS AT DALLAS

[Based on the slides of Stuart Russell and Bryan Pardo]

Sequential Data

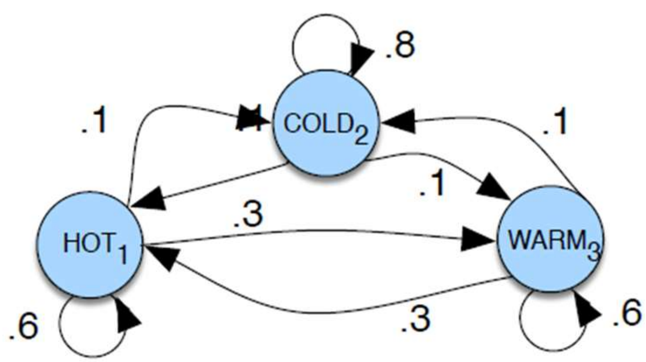
- So far, we have focused primarily on sets of data points that were assumed to be independent and identically distributed.

- However, for many applications, we want to reason about a sequence of observations
  - Speech Recognition
  - Robot Localization
  - Financial Market Analysis
  - Medical Monitoring
  - Weather Forecasting

- Need to introduce time (or space) into our models.

Markov Chain

- A Markov chain is a model that tells us something about the **probabilities of sequences of random variables**, ***states***, each of which can take on values from some set.
- These sets can be expressed by symbols representing anything, like the weather.
- Model is usually represented by a *weighted finite-state automaton*, where the input sequence uniquely determines which states the automaton will go through.
- Example:



Markov Chain Model

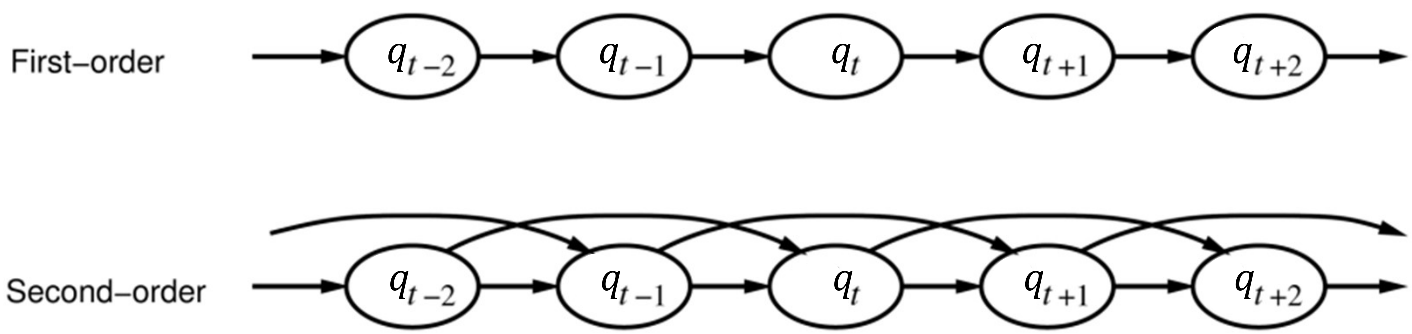
- **States**:
  - $Q = q_1, q_2 \ldots q_N$; the state at time t is q_t
- **Transition Probabilities**:
  - $A = a_{01}, a_{02}, \ldots, a_{n1}, \ldots, a_{nn}$
  - Each a_{ij} represents the probability of transitioning from state i to state j
  - The set of these is the transition probability matrix A
  - $\sum_{j=1}^{n} a_{ij} = 1 \ \forall i$
- **Initial Probability Distribution:**
  - $\pi = \pi_0, \pi_1, \ldots, \pi_N$
  - π_i is the probability that the Markov chain will start in state i.
  - Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also,
  - $\sum_{i=1}^{n} \pi_i = 1$

Markov Chain Model

- **Markov Assumption:** Current state only depends on bounded subset of previous state(s).

First order Markov: $P(q_i|q_1, \ldots, q_{i-1}) = P(q_i|q_{i-1})$

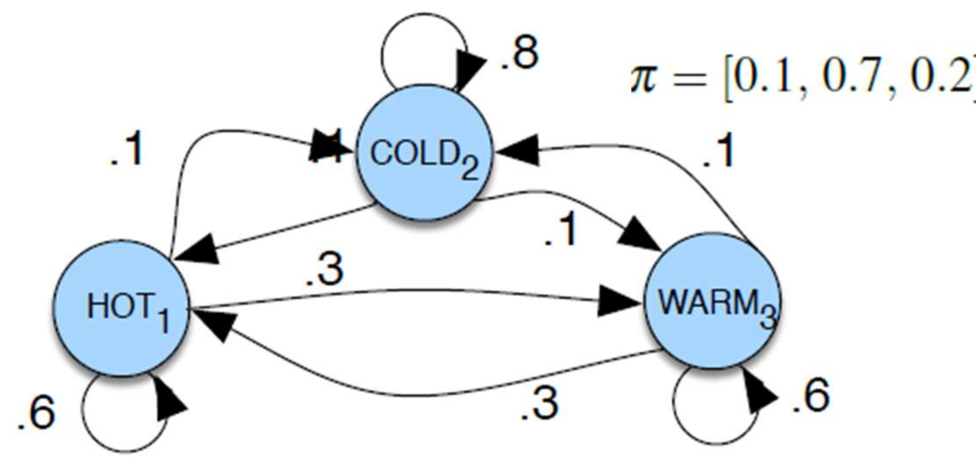
Second order Markov: $P(q_i|q_1, \ldots, q_{i-1}) = P(q_i|q_{i-1}, q_{i-2})$



Markov Chain Model

- Markov chains are useful when we need to compute the probabilities for a sequence of events that are observable.
- Markov chain for assigning a probability to a sequence of weather events: HOT, COLD, and WARM.

- States are represented by the nodes
- Transitions (and their probabilities) are represented by the edges.

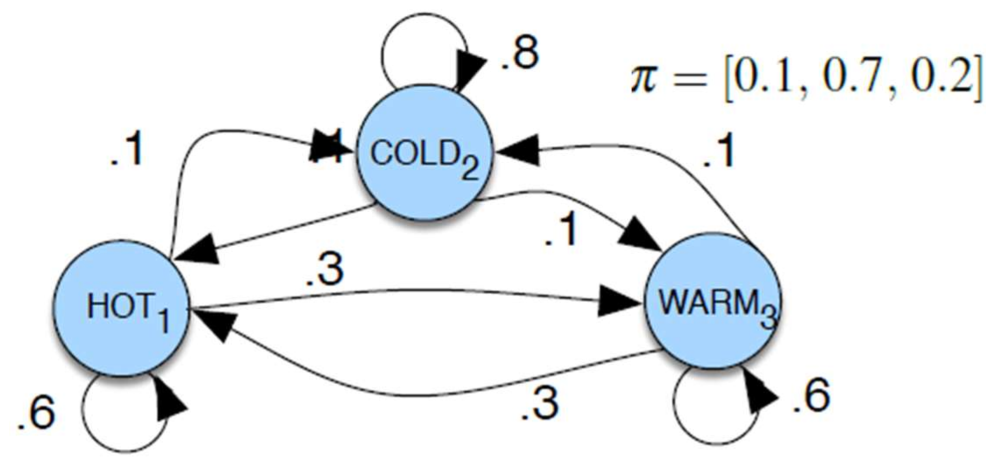


Markov Chain Model

Examples:

$P(WARM, WARM, WARM, WARM) = P(3,3,3,3) = \pi_3 a_{33} a_{33} a_{33} = 0.2 \times 0.6 \times 0.6 \times 0.6 = 0.0432$
$P(HOT, HOT, WARM, COLD) = P(1,1,3,2) = \pi_1 a_{11} a_{13} a_{32} = 0.1 \times 0.6 \times 0.3 \times 0.1 = 0.0018$
$P(COLD, COLD, WARM, COLD) = P(2,2,3,2) = \pi_2 a_{22} a_{23} a_{32} = 0.7 \times 0.8 \times 0.1 \times 0.1 = 0.0056$

- States are represented by the nodes
- Transitions (and their probabilities) are represented by the edges.



Hidden Markov Models

What if we do not have access to the states?

- In many cases, the events we are interested in are **hidden**: we don't observe them *directly*.
- **Example:**
  - **Hidden States:** weather condition (e.g., Sunny, Rainy, Cloudy). We cannot directly see or measure the weather in this context — we only infer it.
  - **Observations:** the activities of people that you can see, such as:
    - People carrying umbrellas
    - People wearing sunglasses
    - People wearing coats
  - **Transition Probabilities:** the likelihood of the weather state changing from one to another
  - **Observation Likelihoods:** the likelihood of seeing an observation given a weather state (e.g., people carrying umbrellas are more likely when it is Rainy).

Hidden Markov Models

- **States**:
  - $Q \; = \; q_1, q_2 \dots q_N$; the state at time t is q_t
- **Observations:**
  - $O = o_1, o_2, \dots, o_T$
- **Transition Probabilities**:
  - Transitional probabilities matrix $A \; = \{a_{ij}\}$
  - $a_{ij} = p(q_t = j \,|q_{t-1} = i) \quad 1 \leq i, j \leq N$
- **Observation Likelihoods or Emission Probabilities:**
  - Output probability matrix $B = \{b_i(o_t)\}$
  - $b_i(o_t) = p(o_t |\, q_t = i)$
- **Initial Probability Distribution:**
  - $\pi_i = P(q_1 = i) \quad\quad 1 \leq i \leq N$

Hidden Markov Models

A first-order hidden Markov model instantiates two simplifying assumptions

- The probability of a particular state depends only on the previous state:

$$\textbf{Markov Assumption:} \quad P(q_i|q_1...q_{i-1}) = P(q_i|q_{i-1})$$

- The probability of an output observation o_i depends only on the state that produced the observation q_i and not on any other states or any other observations:

$$\textbf{Output Independence:} \quad P(o_i|q_1 \dots q_i, \dots, q_T, o_1, \dots, o_i, \dots, o_T) = P(o_i|q_i)$$

Hidden Markov Models

Example:

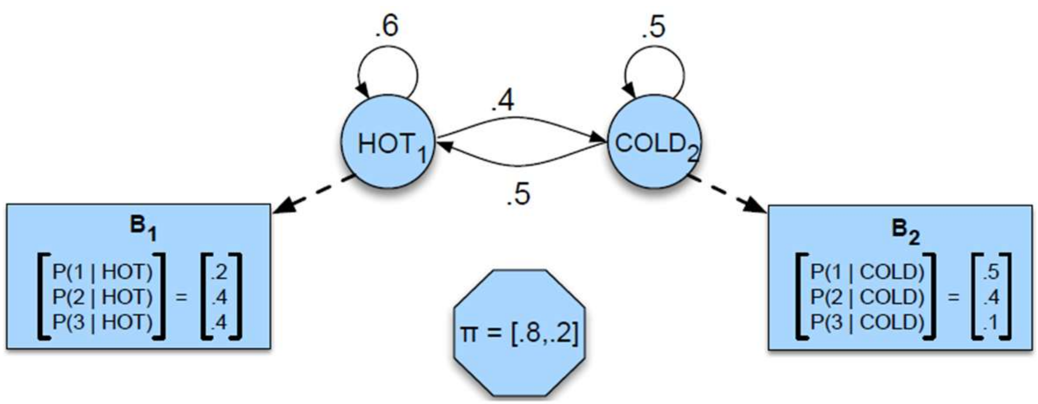
- Imagine that you are a climatologist in the year 2,799 studying the history of global warming.
- You cannot find any records of the weather in Baltimore, Maryland, for the summer of 2020.
- But you do find Jason Eisner's diary, which lists how many ice creams Jason ate every day that summer.
- Our goal is to use these observations to estimate the temperature every day.
- We'll simplify this weather task by assuming there are only two kinds of days: cold (C) and hot (H).

Example from Eisner, J. (2002). An interactive spreadsheet for teaching the forward-backward algorithm.

Hidden Markov Models

Example:

- Given a sequence of observations O (each an integer representing the number of ice creams eaten on a given day) find the 'hidden' sequence Q of weather states (H or C) which caused Jason to eat the ice cream.



Hidden Markov Models

Three Basic Problems for HMM

Problem 1 (Likelihood):	Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O\|\lambda)$.
Problem 2 (Decoding):	Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q.
Problem 3 (Learning):	Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

Hidden Markov Models

Likelihood Computation

Computing Likelihood: Given an $HMM\ \lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O|\lambda)$.

Example: given the ice-cream eating HMM from previous example, what is the probability of the sequence $O = 3, 1, 3$?

Again: Observation sequence O is given, but we don't know the states Q (hot or cold).

Hidden Markov Models

Likelihood Computation: Brute Force

- To compute P($o_1, o_2, \dots, o_n$) using brute force approach, we would have to sum out (marginalize) over all possible values of state variables ($q_1, q_2, \dots, q_n$).
- Each q_i can be assigned one of K states (e.g., hot, cold, etc.)
- Will have to take all possible combinations of these variables to sum out:

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

where $P(O,Q) = P(O|Q) \text{x}\, P(Q) = \prod_{i=1}^{n} P\big(o_{i|} q_i\big) \text{x} \prod_{i=1}^{n} P(q_i | q_{i-1})$

Hidden Markov Models

Likelihood Computation: Brute Force

- To compute P($o_1, o_2, \ldots, o_n$) using brute force approach, we would have to sum out (marginalize) over all possible values of state variables ($q_1, q_2, \ldots, q_n$).
- Each q_i can be assigned one of K states (e.g., hot, cold, etc.)
- Will have to take all possible combinations of these variables to sum out:

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

where $$P(O,Q) = P(O|Q) \text{x}\, P(Q) = \prod_{i=1}^{n} P(o_{i|}q_i) \text{x} \prod_{i=1}^{n} P(q_i|q_{i-1})$$

For an HMM with N hidden states and an observation sequence of T observations, **there are N^T possible hidden sequences.**

Hidden Markov Models

Likelihood Computation: The Forward Algorithm

- Uses Dynamic Programing to perform uses $O(N^2T)$ operations instead of N^T
- $\alpha_t(j)$ is the probability of being in state j after seeing the first t observations

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

- **Idea:** Compute forward probability $\alpha_t(j)$ recursively over t:

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

Remember: $a_{ij} = p(q_t = j \mid q_{t-1} = i) \quad and \ b_i(o_t) = p(o_t \mid q_t = i)$

Hidden Markov Models

Likelihood Computation: The Forward Algorithm

1. Initialization:

$$\alpha_1(j) \;=\; \pi_j b_j(o_1) \;\; 1 \leq j \leq N$$

2. Recursion:

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t); \quad 1 \leq j \leq N, 1 < t \leq T$$

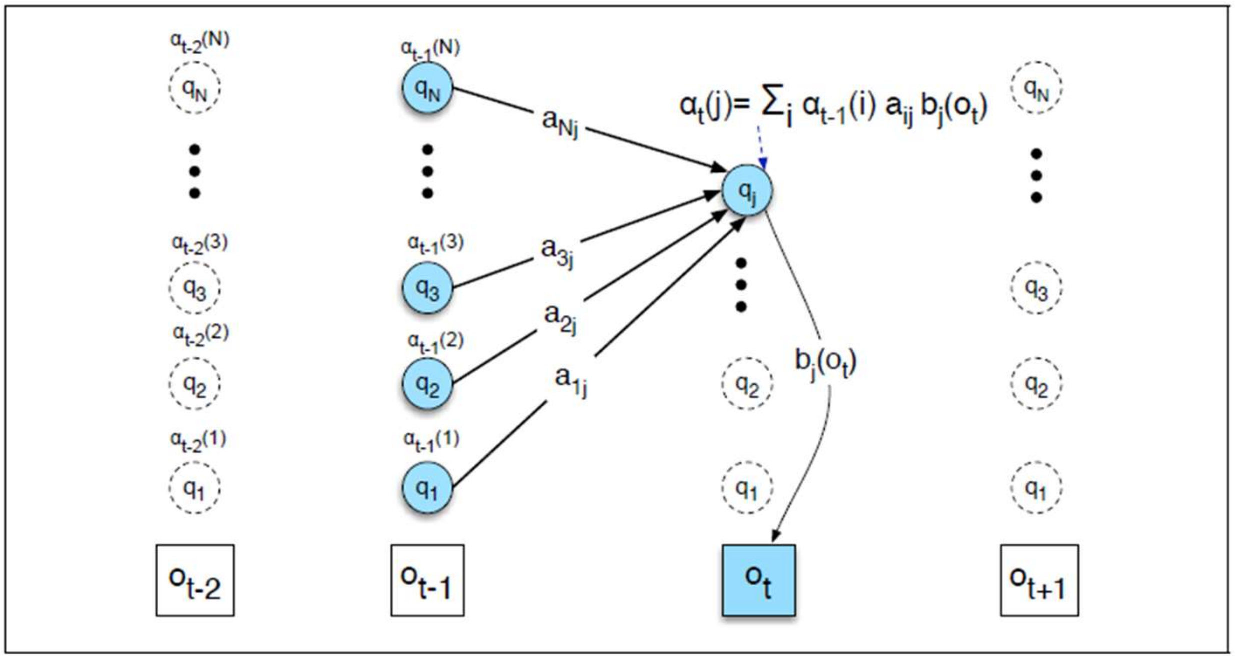
3. Termination:

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

Remember: $\alpha_t(j)$ is the probability of being in state j after seeing the first t observations.

Hidden Markov Models

Likelihood Computation: The Forward Algorithm

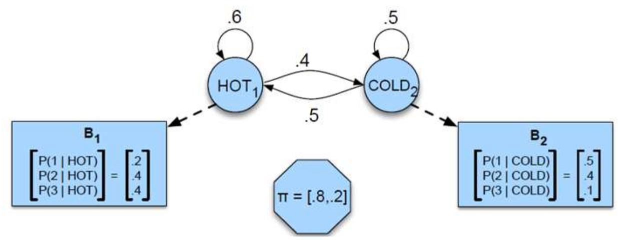


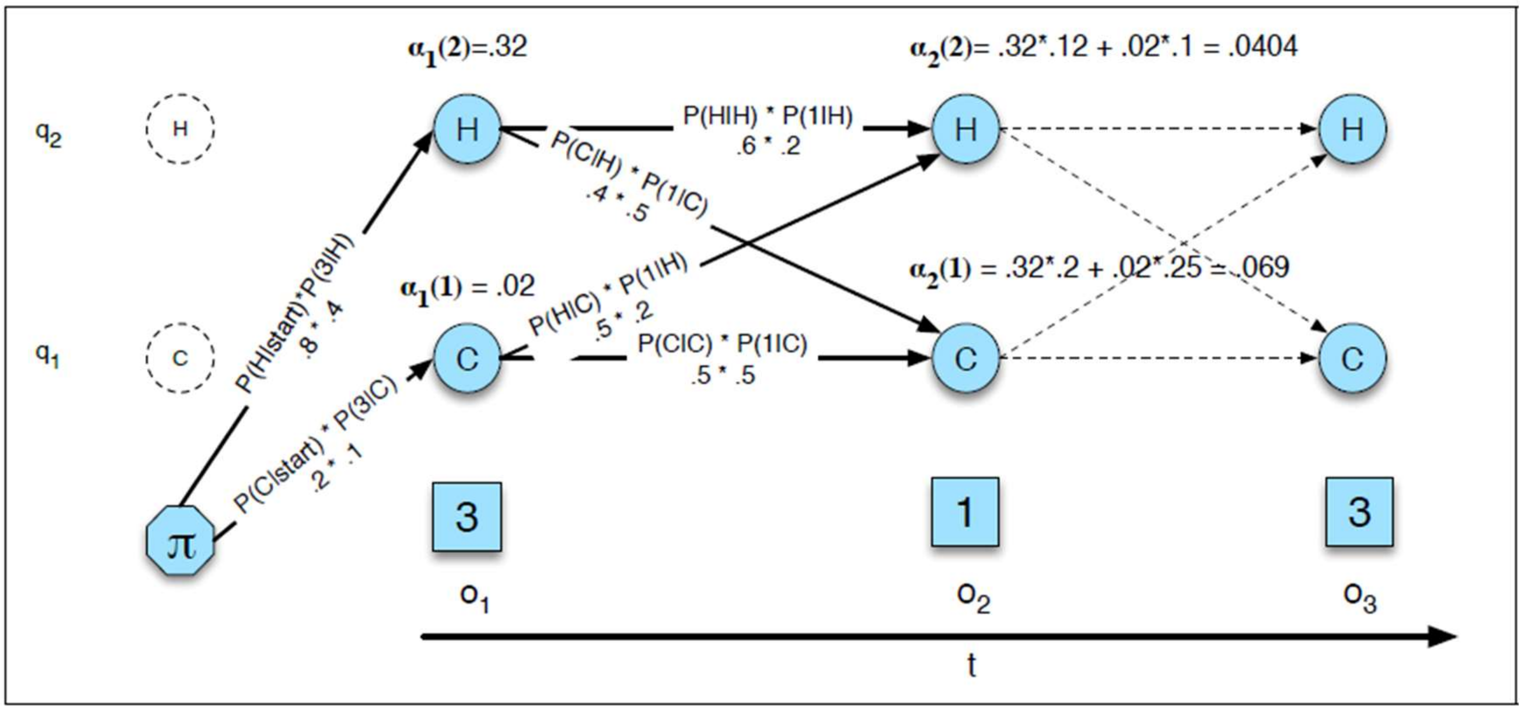
$\alpha_t(j)$ is computed by summing all the previous values α_{t-1} weighted by their transition probabilities a and multiplying by the observation probability $b_j(o_t)$.

- Hidden states are in circles
- Observations in squares.
- Shaded nodes are included in the probability computation for $\alpha_t(j)$.

Hidden Markov Models

Likelihood Computation: Ice Cream Example





Hidden Markov Models

The Forward Algorithm – Example

Given:

Transition Matrix A		***j***	
		S1	**S2**
i	**S1**	0.6	0.4
	S2	0.3	0.7

Initial States	π
S1	0.8
S2	0.2

Emission Probs B		***Possible Observations***		
		R	**W**	**B**
States	**S1**	0.3	0.4	0.3
	S2	0.4	0.3	0.3

Find the probability of O= R,W

Hidden Markov Models

The Forward Algorithm – Example

Find the probability of O= R,W

Initialization (t=1):

$$\alpha_1(S1) = \pi_{S1} b_{S1}(R) = 0.8 \times 0.3 = 0.24$$
$$\alpha_1(S2) = \pi_{S2} b_{S2}(R) = 0.2 \times 0.4 = 0.08$$

Then computing t=2

$$\alpha_2(S1) = \sum_{i=1}^{N} \alpha_1(i) a_{i,S1} b_{S1}(o_2) = \alpha_1(S1) a_{S1,S1} b_{S1}(W) + \alpha_1(S2) a_{S2,S1} b_{S1}(W)$$
$$= 0.24 \times 0.6 \times 0.4 + 0.08 \times 0.3 \times 0.4 = 0.0672$$

$$\alpha_2(S2) = \sum_{i=1}^{N} \alpha_1(i) a_{i,S2} b_{S2}(o_2) = \alpha_1(S1) a_{S1,S2} b_{S2}(W) + \alpha_1(S2) a_{S2,S2} b_{S2}(W)$$
$$= 0.24 \times 0.4 \times 0.3 + 0.08 \times 0.7 \times 0.3 = 0.0456$$

Emission Probs B		Possible Observations		
		R	W	B
States	S1	0.3	0.4	0.3
	S2	0.4	0.3	0.3

Transition Matrix A		*j*	
		S1	S2
i	S1	0.6	0.4
	S2	0.3	0.7

Initial States	π
S1	0.8
S2	0.2

Hidden Markov Models

The Forward Algorithm – Example

Find the probability of O= R,W

Termination step:

$$P(O|\lambda)=\sum_{i=1}^{N}\alpha_T(i) = \alpha_2(S1) + \alpha_2(S2) = 0.0672 + 0.0456 = 0.1128$$

Emission Probs B		*Possible Observations*		
		R	W	B
States	S1	0.3	0.4	0.3
	S2	0.4	0.3	0.3

Transition Matrix A		*j*	
		S1	S2
i	S1	0.6	0.4
	S2	0.3	0.7

Initial States	π
S1	0.8
S2	0.2

Hidden Markov Models

Decoding Computation

Given an $HMM\ \lambda = (A, B)$ and a sequence of observations $O = o_1, o_2, \dots, o_T$, find the most probable sequence of states $Q = q_1, q_2, \dots, q_T$.

- **Brute force approach:** We could just enumerate all paths given the input and use the model to assign probabilities to each.
  - Exponentially large number of state sequences.
- **Instead:** Viterbi Algorithm!

Hidden Markov Models

Decoding Computation: Viterbi Algorithm

Viterbi algorithm computes a trellis using Dynamic Programming (again!!)

The value $v_t(j)$ is computed by recursively taking the **most probable path that could lead us to state j in time step t**.

$$v_t(j) = \max_{q_1,\dots,q_{t-1}} P(q_1 \dots q_{t-1}, o_1, o_2 \dots o_t, q_t = j | \lambda)$$

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t)$$

Where $v_{t-1}(j)$ is the previous Viterbi path probability from the previous time step

Hidden Markov Models

Decoding Computation: Viterbi Algorithm

1. **Initialization:**

$$v_1(j) = \pi_j b_j(o_1) \qquad 1 \leq j \leq N$$
$$bt_1(j) = 0 \qquad 1 \leq j \leq N$$

2. **Recursion**

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t); \quad 1 \leq j \leq N, 1 < t \leq T$$
$$bt_t(j) = \operatorname*{argmax}_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t); \quad 1 \leq j \leq N, 1 < t \leq T$$

3. **Termination:**

$$\text{The best score:} \quad P* = \max_{i=1}^{N} v_T(i)$$
$$\text{The start of backtrace:} \quad q_T* = \operatorname*{argmax}_{i=1}^{N} v_T(i)$$

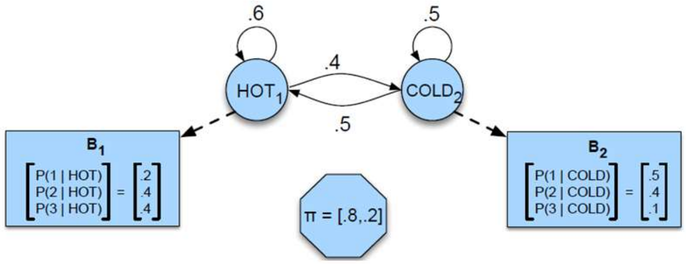
$v_{t-1}(i)$	the **previous Viterbi path probability** from the previous time step
a_{ij}	the **transition probability** from previous state q_i to current state q_j
$b_j(o_t)$	the **state observation likelihood** of the observation symbol o_t given the current state j

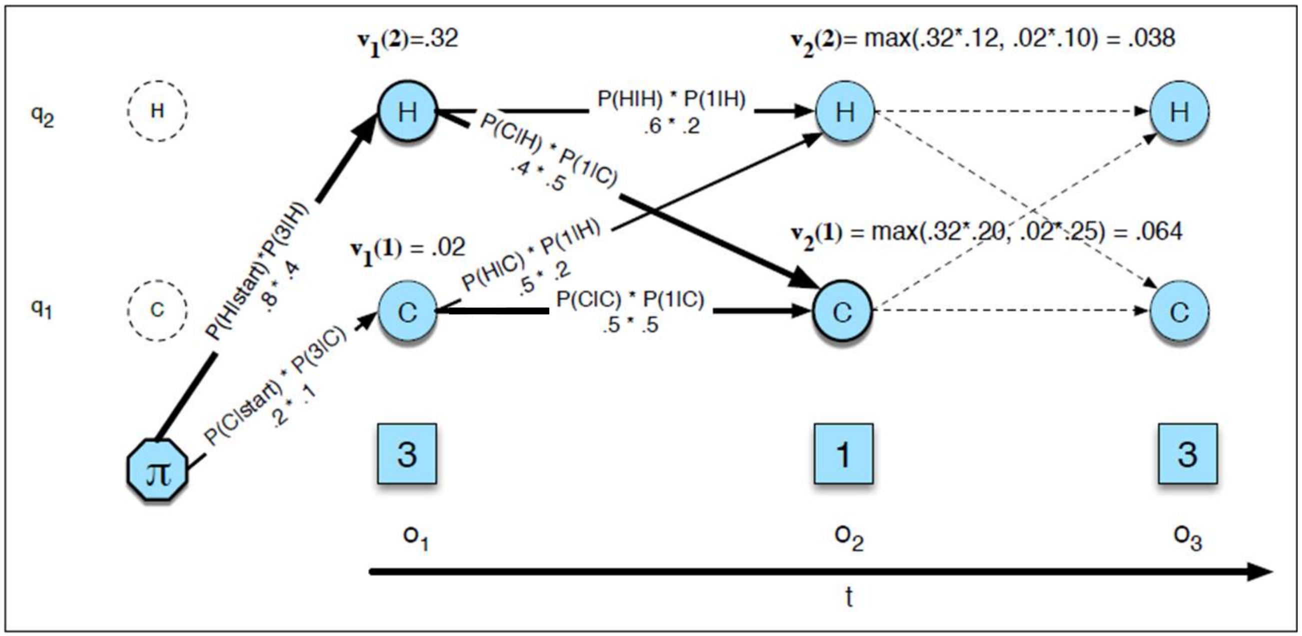
Viterbi recursion, where ***$v_t(j)$*** computes the probability of the most likely sequence of hidden states ending in state j at time t.

Use ***bt_t*** to store the states that maximize the probabilities, so we can later backtrack over the chain. ***$bt_t(j)$*** stores the hidden state from previous time step (t-1) in the most probable path ending in hidden state j.

Hidden Markov Models

The Viterbi Algorithm: Ice Cream Example





Computation of $v_t(j)$ for two states at two time steps.

The computation in each cell follows

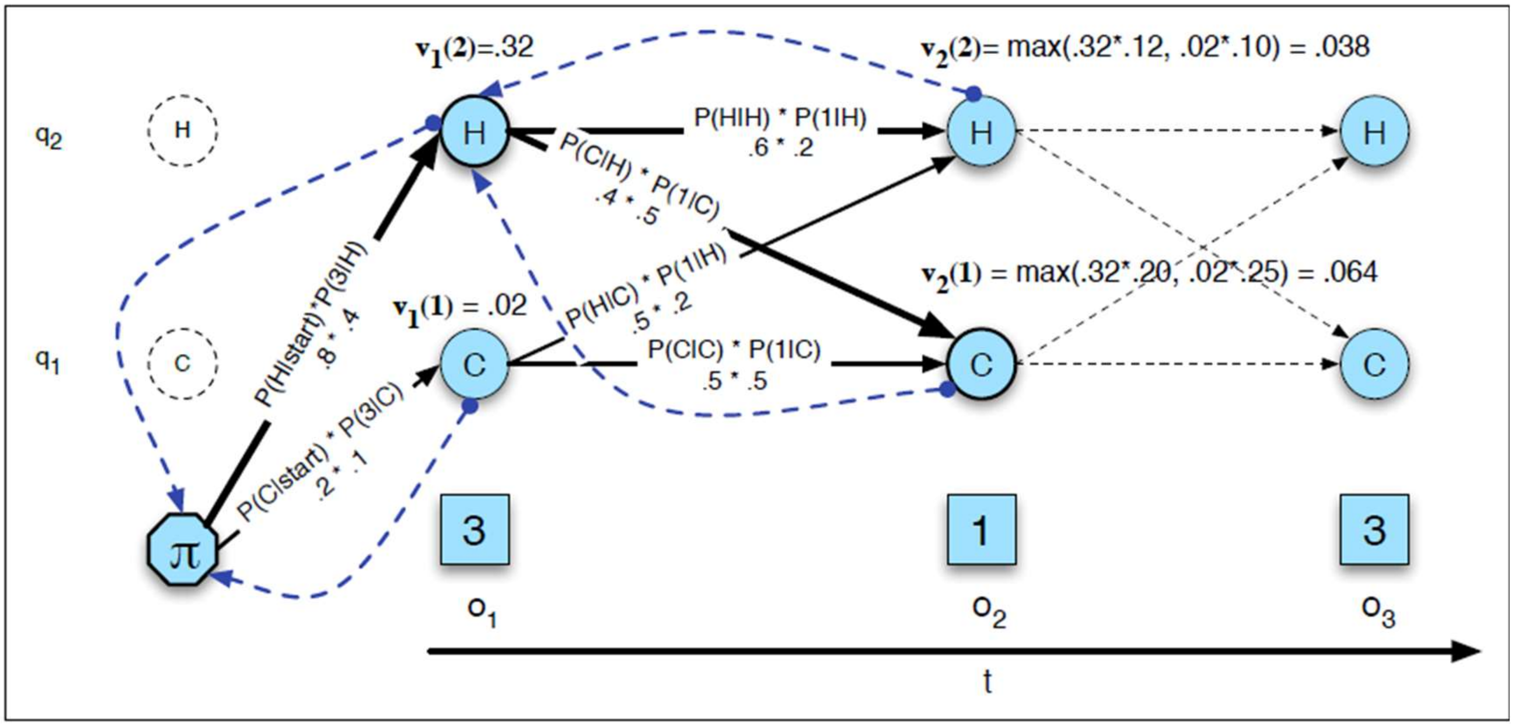
$$v_t(j) = \max_{1 \le i \le N-1} v_{t-1}(i) a_{ij} b_j(o_t)$$

The resulting probability expressed in each cell is $P(q_1, \ldots, q_{t-1}, o_1, \ldots, o_t, q_t = j \mid \lambda)$

- Hidden states are in circles
- Observations in squares

Hidden Markov Models

Decoding Computation: Viterbi Backtrace



Hidden Markov Models

The Viterbi Algorithm: Example

Given the following HMM:
States: *S1, S2* and *S3*
Observations: *a, c, t* and *g*
Transition matrix (A), ***emission probabilities (B)*** *and* ***initial probability distribution (π)*** are given below:

Transition Matrix A		***j***		
		S1	**S2**	**S3**
i	**S1**	0.25	0.50	0.25
	S2	0.25	0.25	0.50
	S3	0.50	0.50	0.0

Initial States	π
S1	0.25
S2	0.50
S3	0.25

Emission Probs B		***Possible Observations***			
		a	**c**	**t**	**g**
States	**S1**	1.00	0.00	0.00	0.00
	S2	0.25	0.50	0.00	0.25
	S3	0.25	0.25	0.25	0.25

Hidden Markov Models

The Viterbi Algorithm: Example

You observe O={C, C, T}. What are the most likely states?

Initialization:

$$v_1(j) = \pi_j b_j(o_1)$$

	C
S1	$\frac{1}{4} * 0 = 0$
S2	$\frac{1}{2} * \frac{1}{2} = \frac{1}{4}$
S3	$\frac{1}{4} * \frac{1}{4} = \frac{1}{16}$

Initial States	π
S1	0.25
S2	0.50
S3	0.25

Emission Probs B		*Possible Observations*			
		a	**c**	**t**	**g**
States	**S1**	1.00	0.00	0.00	0.00
	S2	0.25	0.50	0.00	0.25
	S3	0.25	0.25	0.25	0.25

Hidden Markov Models

The Viterbi Algorithm: Example

You observe O={C, C, T}. What are the most likely states?

Recursion:

$$v_t(j) = \max_{i=1} v_{t-1}(i) a_{ij} b_j(o_t) = \boldsymbol{b_j(o_t)} \ \boldsymbol{\max_{i=1} v_{t-1}(i) a_{ij}}$$

	C	C	T
S1	$\frac{1}{4} * 0 = 0$	$0 * \max\{\ldots\} = 0$	$0 * \max\{\ldots\} = 0$
S2	$\frac{1}{2} * \frac{1}{2} = \frac{1}{4}$	$\frac{1}{2} * \max\{0 * \frac{1}{2},\ \frac{1}{4} * \frac{1}{4},\ \frac{1}{16} * \frac{1}{2}\} = \frac{1}{32}$	$0 * \max\{\ldots\} = 0$
S3	$\frac{1}{4} * \frac{1}{4} = \frac{1}{16}$	$\frac{1}{4} * \max\{0 * \frac{1}{4},\ \frac{1}{4} * \frac{1}{2},\ \frac{1}{16} * 0\} = \frac{1}{32}$	$\frac{1}{4} * \max\{0 * \frac{1}{4},\ \frac{1}{32} * \frac{1}{2},\ \frac{1}{32} * 0\} = \frac{1}{256}$

Transition Matrix A		***j***		
		S1	**S2**	**S3**
i	**S1**	0.25	0.50	0.25
	S2	0.25	0.25	0.50
	S3	0.50	0.50	0.0

Emission Probs B		***Possible Observations***			
		a	**c**	**t**	**g**
States	**S1**	1.00	0.00	0.00	0.00
	S2	0.25	0.50	0.00	0.25
	S3	0.25	0.25	0.25	0.25

Hidden Markov Models

The Viterbi Algorithm: Example

You observe O={C, C, T}. What are the most likely states?

Termination:

$$Best\ Score\ P* = \max_{i=1} v_T(i)$$

$$Start\ of\ BackTrace\ \ q_T* = arg\max_{i=1} v_T(i)$$

	C	C	T
S1	¼ * 0 = 0	0 * max{...} = 0	0 * max{...} = 0
S2	½ * ½ = ¼	½ * max{ 0 * ½, ¼ * ¼, $^1/_{16}$ * ½ } = $^1/_{32}$	0 * max{...} = 0
S3	¼ * ¼ = $^1/_{16}$	¼ * max{ 0 * ¼, ¼ * ½, $^1/_{16}$ * 0 } = $^1/_{32}$	¼ * max{ 0 * ¼, $^1/_{32}$ * ½, $^1/_{32}$ * 0 } = $^1/_{256}$

The trace back yields the best path (sequence of hidden states) through the HMM to produce the output:

$$S2 \longrightarrow S2 \longrightarrow S3$$

Hidden Markov Models

The Viterbi Algorithm: Other Example

Given the following HMM:
States: *Healthy (H)* and *Fever (F)*
Observations: *Normal (N), Cold (C)* and *Dizzy (D)*
Transition matrix (A), ***emission probabilities (B)*** *and* ***initial probability distribution (π)*** are given below:

Transition Matrix A		***j***	
		H	**F**
i	**H**	0.7	0.3
	F	0.4	0.6

Initial States	π
H	0.6
F	0.4

Emission Probs B		***Possible Observations***		
		N	**C**	**D**
States	**H**	0.5	0.4	0.1
	F	0.1	0.3	0.6

Hidden Markov Models

The Viterbi Algorithm: Other Example

You observe O={Normal, Cold, Dizzy}. What are the most likely states?

Initialization:

$$v_1(j) = \pi_j b_j(o_1)$$

States	Observations		
	Normal	Cold	Dizzy
Healthy	$0.6 * 0.5 = \mathbf{0.3}$		
Fever	$0.4 * 0.1 = 0.04$		

Initial States	π
H	0.6
F	0.4

Emission Probs B		*Possible Observations*		
		N	C	D
States	H	0.5	0.4	0.1
	F	0.1	0.3	0.6

Hidden Markov Models

The Viterbi Algorithm: Other Example

You observe O={Normal, Cold, Dizzy}. What are the most likely states?

Recursion:

$$v_t(j) = b_j(o_t) \max_{i=1} v_{t-1}(i) a_{ij}$$

Emission Probs B		Possible Observations N	C	D
States	H	0.5	0.4	0.1
	F	0.1	0.3	0.6

Transition Matrix A		*j* H	F
i	H	0.7	0.3
	F	0.4	0.6

States	Observations Normal	Cold	Dizzy
Healthy	$\mathbf{0.3}$	$0.4 * \max\left\{\begin{matrix}\mathbf{0.3 * 0.7,} \\ 0.04 * 0.4\end{matrix}\right\} = \mathbf{0.084}$	$0.1 * \max\left\{\begin{matrix}\mathbf{0.084 * 0.7,} \\ 0.027 * 0.4\end{matrix}\right\} = 0.0059$
Fever	0.04	$0.3 * \max\left\{\begin{matrix}\mathbf{0.3 * 0.3,} \\ 0.04 * 0.6\end{matrix}\right\} = 0.027$	$0.6 * \max\left\{\begin{matrix}\mathbf{0.084 * 0.3,} \\ 0.027 * 0.6\end{matrix}\right\} = \mathbf{0.0151}$

Hidden Markov Models

The Viterbi Algorithm: Other Example

You observe O={Normal, Cold, Dizzy}. What are the most likely states?

Termination: $Best\ Score\ P* = \max_{i=1} v_T(i)$

$Start\ of\ BackTrace\ \ q_T* = arg \max_{i=1} v_T(i)$

The trace back yields the best path (sequence of hidden states) through the HMM to produce the output:

$\boldsymbol{Healthy \longrightarrow Healthy \longrightarrow Fever}$

States	Observations		
	Normal	Cold	Dizzy
Healthy	$\mathbf{0.3}$	$0.4 * \max\begin{Bmatrix}\mathbf{0.3 * 0.7,} \\ 0.04 * 0.4\end{Bmatrix} = \mathbf{0.084}$	$0.1 * \max\begin{Bmatrix}\mathbf{0.084 * 0.7,} \\ 0.027 * 0.4\end{Bmatrix} = 0.0059$
Fever	0.04	$0.3 * \max\begin{Bmatrix}\mathbf{0.3 * 0.3,} \\ 0.04 * 0.6\end{Bmatrix} = 0.027$	$0.6 * \max\begin{Bmatrix}\mathbf{0.084 * 0.3,} \\ 0.027 * 0.6\end{Bmatrix} = \mathbf{0.0151}$

Hidden Markov Models

HMM Training: Baum-Welch Algorithm

Given an observation sequence O and the set of possible states in the HMM, learn the HMM parameters A and B.

- **Expectation-Maximization (EM) method:** estimates HMM parameters to maximize the likelihood of observed data sequences.
- **Algorithm**:
  Expectation (E-step): Uses Forward-Backward algorithm to compute expected counts of transitions and emissions based on the current parameters.
  Maximization (M-step): Updates A, B, and π using the expected counts to maximize the likelihood of the observed data.
  Repeat Until Converge (when changes in the log-likelihood are below a threshold)

Readings

- **Pattern Recognition and Machine Learning, by Bishop – Sections 13.1 and 13.2**
- **Speech and Language Processing, by Daniel Jurafsky – Chapter A.1**