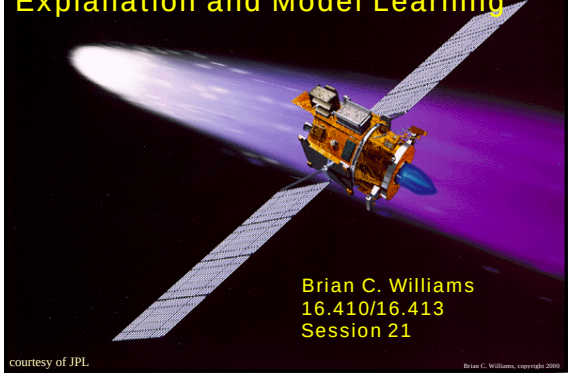
Hidden Markov Models: Explanation and Model Learning



Brian C. Williams
16.410/16.413
Session 21

courtesy of JPL

Reading Assignments

AIMA (Russell and Norvig)

- Ch 15.1-.3, 20.3 State Estimation and Hidden Markov Models

From last Monday:

- Ch 13 Review of Probabilities
- Ch 14.1-4 Probabilistic Reasoning

Outline

- Review
- Explanation and Learning in Statistical Natural Language
  - Decoding using the Viterbi Algorithm
  - Evaluation via Forward and Backward Algorithms.
  - Model learning via the Baum-Welch Algorithm.

HMM Estimation is Pervasive

Engineering Operations

Dialogue Management

Robot Localization

9/13/00 copyright Brian Williams, 2000 4

Courtesy of Kanna Rajan, NASA Ames. Used with permission.

Posterior Probability, after Observations $X_{1,n} = x_{1,n}$

$$P(M) = \prod_{M_i \in M} P(M_i)$$

Assume:
- Apriori mode independence.
- Consistent obs equally likely

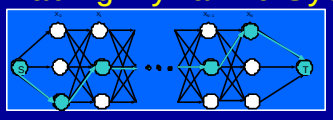
$$P(M \mid x_{1,n}) = a\, P(x_n \mid M)\, P(M \mid x_{1,n-1})$$

$P(x_i \mid M)$ is estimated using model, F, according to:

- **If** previous observations $X_{1,i-1} = x_{1,i-1}$, M and F entails $X_i = x_i$
  Then $P(x_i \mid M) = 1$
- **If** previous observations $X_{1,i-1} = x_{1,i-1}$, M and F entails X_i ? v_i
  Then $P(x_i \mid c) = 0$
- **Otherwise**, Assume all consistent assignments to X_i are equally likely observations:
  let $D_{ci} = \{x_{ci} \in D_{xi} \mid c, F$ is consistent with $X_i = x_c\}$
  Then $P(x_i \mid M) = 1/|D_{ci}|$

9/13/00 copyright Brian Williams, 2000 5

Estimating Dynamic Systems

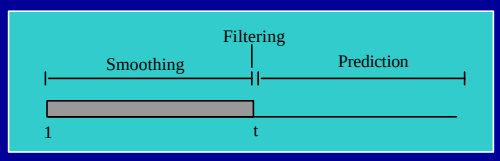


Given a sequence of observations and commands:

- What is the likelihood of a particular state?
  - ⇨ **Belief State Update**: (filtering and smoothing)
- What is the most likely sequence of states that got me here?
  - ⇨ Decoding: (Viterbi Algorithm)
- What is the most likely sequence of observations generated?
  - ⇨ Evaluation/Prediction:
- What HMM most likely generated these observations?
  - ⇨ Learning: (Baum-Welch Algorithm, Expectation-Maximization Algorithm)

9/13/00 copyright Brian Williams, 2000 6

What is the likelihood of a state?



- Filtering
  - Probabilities of current states
- Prediction
  - Probabilities of future states
- Smoothing
  - Probabilities of past states

Notation

- $\mathbf{S}^{t+1}$: set of hidden variables in the t+1 time slice
- $\mathbf{s}^{t+1}$: set of values for those hidden variables at t+1
- **$\mathbf{x}^{t+1}$: set of observations at time t+1**
- **$\mathbf{x}^{1:t}$: set of observations from all times from 1 to t**
- **a: normalization constant**

Hidden Markov Models

- Finite States S, Actions A & Observations Ω
- State transition function
  $T(S^i, A^i, S^{i+1}) \equiv P(S^{i+1} | S^i, A^i)$
- Observation function
  $O(S^i, \Omega^i) \equiv P(\Omega^i | S^i)$
- Initial state distribution
  $\Theta(S)$: $P(S^1)$

Notation:
$\Pi(S)$ denotes all subsets of S

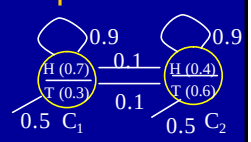
Markov Assumptions

Given a distribution over the current state, the future states and current and future observations are independent of the past.

- **First-order Markov process**
  - $P(S^t|S^{0:t-1}) = P(S^t|S^{t-1})$
- **Markov assumption of evidence**
  - $P(X^t|S^{0:t},X^{0:t-1}) = P(X^t|S^t)$

Belief Update Example

$$b(S^{i+1}) = a\,O(x^{i+1}, S^{i+1}) \sum_{s' \in S^i} T(S^i, a^i, S^{i+1}) b(S^i)$$



Observed sequence:

H T H H H H T H

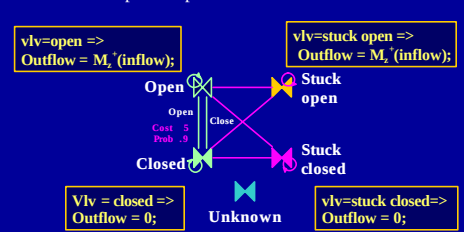
C_1 0.5 a x 0.7 x [0.9 x 0.5 + 0.1 x 0.5] = 0.35a = 0.64

C_2 0.5 a x 0.4 x [0.1 x 0.5 + 0.9 x 0.5] = 0.20a = 0.36

$\sum_{s_i \in S} P(s_i) = 1$, hence 0.35a + 0.20a = 1 ⇨ a = 1.82

Diagnosing Dynamic Systems: Via Probabilistic Constraint Automata

- Devices modes
- Probabilistic transitions between modes
- State constraints for each mode
- One automata per component



Outline

- Review
- Explanation and Learning in Statistical Natural Language
  - Decoding using the Viterbi Algorithm
  - Evaluation via Forward and Backward Algorithms
  - Model learning via the Baum-Welch Algorithm

Variant on a Hidden Markov Model

A HMM is defined as $<s_1,S,W,E>$

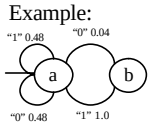
- S is the set of states,
- $s_1 \in S$ is the single start state,
- W is the set of observable symbols,
- E is the set of "emitting" transitions.
  - A transition is a four tuple such as $<s_2,$"had", $s_3, 0.3>$ denoting:

$$P(s_2 \xrightarrow{"had"} s_3) = 0.3 \equiv P(s_3, "had" \mid s_2)$$

Differences from earlier HMM:
- Observations are "words"
- Transitions "emit" observations
- A unique start state

E combines the transition and observation function in the previous HMM def.

Example:



s_1 = a

S = { a, b }

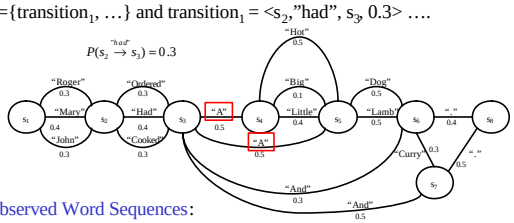
W = { 0, 1 }

E = { <a, "1", a, 0.48>
<a, "0", a, 0.48>
<a, "0", b, 0.04>
<b, "1", a, 1.0> }

Sentence Parsing Example

HMM = $<S,s_1,W,E>$ where $S=\{s_1,s_2,s_3,s_4,s_5,s_6,s_7,s_8\}$; W={"Roger", …}; E={$\text{transition}_1$, …} and $\text{transition}_1 = <s_2,$"had", s_3, 0.3> ….

$$P(s_2 \xrightarrow{"had"} s_3) = 0.3$$



Observed Word Sequences:

S_1: Mary had a little Lamb and a big dog.
S_2: Roger ordered a lamb curry and a hot dog.
S_3: John cooked a hot dog curry.

$P(S_3)$=0.3*0.3*0.5*0.5*0.5*0.3*0.5=0.003375

Problems and Algorithms

Decoding:

Given a sequence of observations, what is the most likely sequence of hidden states?

- Solution: Viterbi algorithm

Evaluation: What is the probability of a given sequence of observations?

- Solution: forward/backward algorithm
  - (These are used in the learning algorithm).

Learning:

Given a sequence of observations, what HMM transition probabilities maximize the likelihood of the sequence?

- Solution: Baum-Welch algorithm (form of Expectation-Maximization)

Outline

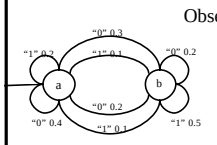
- Overview
- Belief State Update
- Explanation and Learning in Statistical Natural Language
  - Decoding using the Viterbi Algorithm
  - Evaluation via Forward and Backward Algorithms
  - Model learning via the Baum-Welch Algorithm.

HMM Decoding

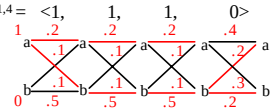
Finding the most likely state trajectory

Problem: Given HMM = <S, s_1,W,E>, and observation sequence $w^{1:t-1}$, find the most likely state sequence (denoted $\sigma_i(t)$), ending in s_i at time t:

$$s_i(t) = \arg\max_{s^{1:t} \text{ s.t. } s^t = s_i} P(s^{1:t} \mid w^{1:t-1})$$



Observe $w^{1,4}$ = <1, 1, 1, 0>



σ_a^5 = <a, b, b, b, a>

Probability of the Most Likely State Trajectory

Probability of the most likely sequence $\sigma_i(t)$ ending in s_i given observations $w^{1,t}$:

$$P(s_i(t+1) \mid w^{1:t}) = \frac{P(s_i(t+1), w^{1:t})}{P(w^{1:t})}$$

$$= a \max_{s_j \in S} P(s_i^{t+1}, w^t \mid s_j^t) P(s^{1:t-1}, s_j^t \mid w^{1:t-1})$$

$$P(s_i(t+1) \mid w^{1:t}) = a \max_{s_j \in S} P(s_i^{t+1}, w^t \mid s_j^t) P(s_j(t) \mid w^{1:t-1})$$

$$P(s_i(1)) = 1, \text{ if } s_i^1 = s_1, \text{ and 0, otherwise.}$$

Viterbi Algorithm

Problem: Compute

$$s_i(t) = \arg \max_{s^{1:t} \text{ s.t. } s^t = s_i} P(s^{1:t} \mid w^{1:t-1})$$

Solution: For n from 1 to t,

- Compute the most likely paths of length n that end at each $s_k \in S$.
- Extend to the most likely paths of length n+1 that end at each $s_i \in S$

$$s_i(1) = \langle s_i \rangle$$

$$s_i(t+1) = s_{j_{\max}}(t) \circ s_i,$$

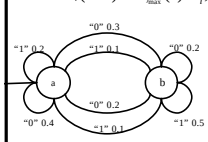
$$\text{where } j_{\max} = \arg\max_{k=1}^{|S|} P(s_k(t) \mid w^{1:t-1}) P(s_k \xrightarrow{w^t} s_i)$$

Notation: $\langle s_1, \ldots s_n \rangle \circ b \equiv \langle s_1, \ldots s_n, b \rangle$

Example: Viterbi Algorithm

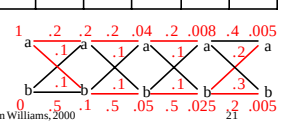
$$s_i(1) = \langle s_i \rangle$$

$$s_i(t+1) = s_{i_{\max}}(t) \circ s_i, \text{where } j_{\max} = \arg\max_{k=1}^{|S|} P(s_k(t)) P(s_k \xrightarrow{w^t} s_i)$$



States\ Obs		Start	1	11	111	1110
a	Sequence	a	aa	aaa	aaaa	abbba
	Probability	1.0	0.2	0.04	0.008	0.005
b	Sequence	b	ab	abb	abbb	abbbb
	Probability	0.0	0.1	0.05	0.025	0.005

Observe "1110"



Viterbi Pseudo Code

Viterbi(<S,s_1,W,E>, $w^{1:T-1}$, T) // **param** $\alpha_i(t)$ denotes $P(\sigma_i(t)|w^{1:t-1})$

1. begin
2. **for** ($i \leftarrow 1$; $i \leq |S|$; $i \leftarrow i+1$) {
3. initialize $\sigma_i(1) \leftarrow \{s_i\}$, $\alpha_i(1) \leftarrow \{1$ **if** s_i is start state s_1, **else** $0\}\}$
4. **for** ($t \leftarrow 1$; $t < T$; $t \leftarrow t+1$) { // for each obs $w^t \in w^{1,T}$
5. **for** ($i \leftarrow 1$; $i \leq |S|$; $i \leftarrow i+1$) { // each state s_i^{t+1} at time t+1
6. **initialize** $j_{max} \leftarrow 0$, $P_{max} \leftarrow -1$;
7. **for** ($k \leftarrow 1$; $k \leq |S|$; $k \leftarrow k+1$) // compute arg max over s_k^t
8. **if** ($\alpha_k(t) * P(s_i^{t+1}, w^t | s_k^t) > P_{max}$) {
9. $P_{max} \leftarrow \alpha_k(t) * P(s_i^{t+1}, w^t | s_k^t)$, $j_{max} \leftarrow k$ };
10. $\alpha_i(t+1) \leftarrow P_{max}$;
11. $\sigma_i(t+1) \leftarrow \sigma_{jmax}(t) \circ s_i$ } // postpend the next state
12. **return** $\sigma_i(T)$ for $0 < i = |S|$ that maximizes $\alpha_i(T)$
13. end

Outline

- Review
- Explanation and Learning in Statistical Natural Language
  - Decoding using the Viterbi Algorithm
  - Evaluation via Forward and Backward Algorithms
  - Model learning via the Baum-Welch Algorithm
- Appendix: Monitoring and Diagnosis via Probabilistic Constraint Automata

9/13/00 copyright Brian Williams, 2000 23

Probability of Observation Sequence

Forward probability $a_i(t) = P(w^{1:t-1}, s_i^t)$

Similar to belief state, given earlier.

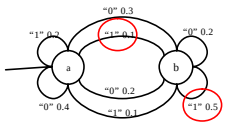
$$a_i(1) = P(w^{1:0}, s_i^1) = \begin{cases} i = 1 \rightarrow 1.0 \\ otherwise \rightarrow 0 \end{cases}$$

$$a_i(t+1) = \sum_{j=1}^{c} P(s_j \xrightarrow{w^t} s_i) a_j(t) = \sum_{j=1}^{c} P(s_j \xrightarrow{w^t} s_i) P(w^{1:t-1}, s_j^t)$$

Observation probability $P(w^{1:t}) = \sum_{i=1}^{|S|} a_i(t+1)$

HMM Evaluation

Observation and Forward probabilities



t	1	2	3	4	5
	ε	1	1	1	0
$\alpha_i(t)$	1.0	**0.2**	0.05	0.017	0.0148
$\alpha_i(t)$	0.0	**0.1**	**0.07**	0.04	0.0131
$P(w^{1:t})$	1.0	0.3	0.12	0.057	**0.0279**

0.2*0.1=0.02
+
0.1*0.5=0.05

"1110"

$$a_i(t+1) = \sum_{j=1}^{c} P(s_j \xrightarrow{w^t} s_i) a_j(t)$$

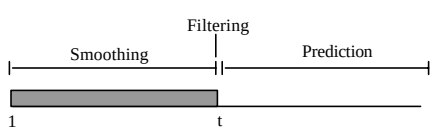
$$P(w^{1:t}) = \sum_{i=1}^{|S|} a_i(t+1)$$

Forward Algorithm Pseudo Code

Forward(<S,s_1,W,E>, $w^{1:T}$) // **param** $\alpha_i(t)$ denotes $P(s_i^t|w^{1:t-1})$

1. **begin**
2. **for** $1 \le i \le |S|$)
3. **initialize** $\alpha_i(1) \leftarrow$ {1 **if** s_i is start state s_1, **else** 0};
4. **for** $1 \le t \le T$ {
5. **for** $1 \le j \le |S|$ {
6. **initialize** $\alpha_j(t+1) \leftarrow 0$;
7. **for** $1 \le i \le |S|$
8. $\alpha_j(t+1) \leftarrow \alpha_j(t+1) + P(s_j, w^t | s_i)*\alpha_i(t)$;
9. }
10. **return** ? $\alpha_i(T+1)$ for all $s_i \in S$
11. **end**

What is the likelihood of a state?



- Filtering
  - Probabilities of current states $P(S^t | w^{1:t})$
- Prediction
  - Probabilities of future states $P(S^k | w^{1:t})$ for k > t
- Smoothing
  - Probabilities of past states $P(S^k | w^{1:t})$ for k < t

Smoothing

$$P(S^k \mid w^{1:t}) = P(S^k \mid w^{1:k}, w^{k+1:t})$$ Divide obs

$$= a\, P(S^k \mid w^{1:k}) P(w^{k+1:t} \mid S^k, w^{1:k})$$ Bayes

$$= a\, P(S^k \mid w^{1:k}) P(w^{k+1:t} \mid S^k)$$ Markov

$$\boxed{P(S^k \mid w^{1:t}) = a\, a_i(k)\, b_i(k)}$$

Backward Probabilities

Backward probability $b_i(t) \equiv P(w^{t:T} \mid s_i^t)$

$\beta_i(t)$ is similar to $\alpha_i(t)$ but starts from the end.

$$b_i(T+1) = P(e \mid s_i^{T+1}) = 1$$

$$b_j(t-1) = \sum_{j=1}^{|S|} P(s_i \xrightarrow{w^{t-1}} s_j) b_j(t) = \sum_{j=1}^{|S|} P(s_i \xrightarrow{w^{t-1}} s_j) P(w^{t:T} \mid s_j^t)$$

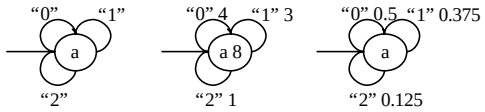
Observation probability $P(w^{1:T}) = b_1(1) = P(w^{1:T} \mid s^1 = s_1^1)$

Outline

- Review
- Explanation and Learning in Statistical Natural Language
  - Decoding using the Viterbi Algorithm
  - Evaluation via Forward and Backward Algorithms
  - Model learning via the Baum-Welch Algorithm

HMM Training
(Baum-Welch Algorithm)

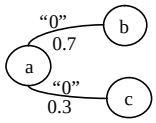
Approach: Given a training sequence $w^{1:T}$, adjust the HMM state transition probabilities to make the observation sequence as likely as possible.



Training Sequence: $w^{1,8}$ = 01010210

Dealing with Hidden States

Intuitively…



When counting transitions "prorate" each transition by its probability.

Problem: States are not known.
Solution: Estimate states from Model.

Problem: Transitions no longer deterministic.
Solution: Compute expected # of transitions.

Problem: Model not known.(chicken &egg)
Solution: Bootstrap the model:
1. Guess a model (transition probabilities).
2. Use model to estimate states.
3. Count estimated transitions to get model.

Expectation-Maximization (Baum-Welch)

1. Guess a set of transition probabilities.
2. while (transition probabilities improving) {
   a. Expectation: Use transition P to estimate states $P(S^t | w^{1:T})$.
   b. Maximization: Estimate new transition probabilities by counting expected # of transitions, given state estimates.}

- "improvement" measured by comparing *cross-entropy* after each iteration:

$$-\frac{1}{n}\sum_{w^{1:T}} P_{M-1}(w^{1:T})\log_2 P_M(w^{1:T})$$

- Terminate when change in cross-entropy is less than some θ.

Estimating The Transition Probability

- $C(s_i, w_k, s_j)$: The expected "count" of transitions $s_i \xrightarrow{w_k} s_j$, during observation sequence $w^{1:T}$:

$$C(s_i \xrightarrow{w_k} s_j) = \sum_{t=1}^{T} a_i(t) P(s_i \xrightarrow{w_k} s_j) b_j(t+1)$$

- P_e: Estimated transition probability for $s_i \xrightarrow{w_k} s_j$ estimated from observation sequence $w^{1:T}$:

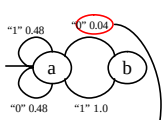
$$P_e(s_i \xrightarrow{w_k} s_j) = \frac{C(s_i \xrightarrow{w_k} s_j)}{\sum_{l=1}^{|S|} \sum_{m=1}^{|W|} C(s_i \xrightarrow{w_m} s_l)}$$

$C(s_i)$: Expected count of transitions out of s_i

Baum-Welch Pseudo Code

```
Baum-Welch (P_new, w^{1:T}, θ)        // P_new ≡ estimated P(s_j, w_k|s_i)
do (                                  // w^{1:T} is a training sequence and θ is a convergence criteria.
  for 1 = i, j = |S|, 1 = k = |W|
    P_old (s_j,w_k|s_i) = P_new (s_j,w_k|s_i);   // remember old probability estimate
  compute α_i(t), β_j(t), for all values of 1 =i = |S| and 1 = t = T;
  for 1 = i, j = |S|, 1 = k = |W|, and 1 = t = T {
    initialize C(s_i, w_k, s_j)← 0;
    for 1 = t = T
      C(s_i, w_k, s_j) ← α_i(t) P_old(s_j,w_k | s_i) β_j(t);
  }
  for 1 = i = |S| {
    initialize C(s_i) ← 0;
    for 1 = j = |S|, 1 = k = |W|
      C(s_i) ← C(s_i) + C(s_i, w_k, s_j);
  for 1 = j = |S|, 1 = k = |W|
    P_new(s_j, w_k | s_i)=C(s_i,w_k, s_j)/C(s_i);
  }
} while (maxChanged(P_new, P_old) > θ)
```

Baum-Welch example



Transition probabilities are initially guessed. Training sequence is "01011"

	1	2	3	4	5	6
	ε	0	1	0	1	1
$\alpha_a(t)$	1	0.48	0.27	0.13	0.072	0.035
$\alpha_b(t)$	0	0.04	0	0.01	0	0

*

	1	2	3	4	5	6
	ε	0	1	0	1	1
$\beta_a(t)$	0.035	0.062	0.13	0.23	0.48	1
$\beta_b(t)$	0	0.13	0	0.28	1	1

	0	1	0	1	1	Total	New P
T(a,0,b)	0.0052	0	0.0052	0	0	0.01	0.06
T(b,1,a)	0	0.0052	0	0.0048	0	0.01	1
T(a,0,a)	0.030	0	0.03	0	0	0.06	0.36
T(a,1,a)	0	0.03	0	0.03	0.035	0.095	0.58

$$\frac{C(a,0,b)}{C(a,0,b)+C(a,0,a)+C(a,1,a)}$$

Notation Summary

Probabilistic transitions:

- written as $P(s_3\text{"had"} \mid s_2) = 0.3$ or as $P(s_2 \xrightarrow{\text{"had"}} s_3) = 0.3$

Observation sequences:

- $w^{1:T}$ denotes the entire sequence of observations.
- ε denotes the empty sequence (no observations).

States S

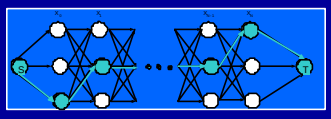
- States are subscripted, $s_i \in S$, where $1 = i = |S|$.
- Superscripts indicate time, for example, s^k is the kth state in a state sequence.

State sequences:

- $\sigma_i(t)$ denotes the most likely sequence of t states that ends in state s_i
- $\sigma_i(t\text{-}1) \circ s_i^t$ concatenates state to the end of the sequence.
- $\alpha_i(t) \equiv a P(s_i^t \mid w^{1:t-1})$ denotes the forward probability at time step t of $s^t = s_i$
- $\beta_i(t) \equiv a P(w^{t:T} \mid s_i^t)$ denotes the backward probability at time step t of $s^t = s_i$

9/13/00 copyright Brian Williams, 2000 37

Estimating Dynamic Systems



Given a sequence of observations and commands:

- What is the likelihood of a particular state?
  - ⇨ **Belief State Update**: (filtering, smoothing, prediction)
- What is the most likely sequence of states that got me here?
  - ⇨ **Decoding**: (Viterbi Algorithm)
- What is the most likely sequence of observations generated?
  - ⇨ **Evaluation**:
- What HMM most likely generated these observations?
  - ⇨ **Learning**: (Baum-Welch Algorithm, Expectation-Maximization Algorithm)

9/13/00 copyright Brian Williams, 2000 38