Hidden Markov Models (HMMs)

ChengXiang Zhai

Department of Computer Science

University of Illinois, Urbana-Champaign

Central Questions to Ask about a LM: “ADMI”

- **Application**: Why do you need a LM? For what purpose?

  → Evaluation metric for a LM

  Topic mining and analysis
  Sequential structure discovery

- **Data:** What kind of data do you want to model?

  → Data set for estimation & evaluation

  Sequence of words

- **Model:** How do you define the model?

  → Assumptions to be made

  Latent state, Markov assumption

- **Inference:** How do you infer/estimate the parameters?

  Viterbi, Baum-Welch

  → Inference/Estimation algorithm

Modeling a Multi-Topic Document

A document with 2 subtopics, thus 2 types of vocabulary

. . .
text mining passage

food nutrition passage

text mining passage

text mining passage

food nutrition passage
. . .

How do we model such a document?

How do we "generate" such a document?

How do we estimate our model?

We've already seen one solution – unigram mixture model + EM
This lecture is about another (better) solution – HMM + EM

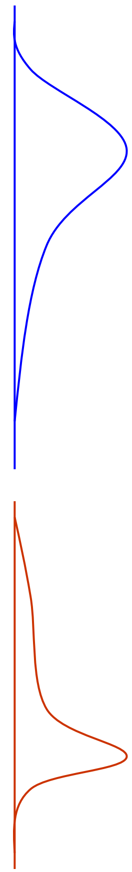
Simple Unigram Mixture Model

Model/topic 1
$p(w|\theta_1)$

…
text 0.2
mining 0.1
assocation 0.01
clustering 0.02
…
food 0.00001
…

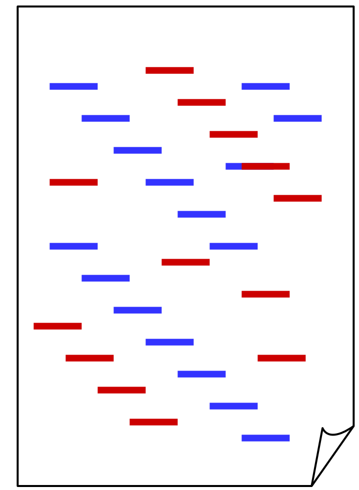
Model/topic 2
$p(w|\theta_2)$

…
food 0.25
nutrition 0.1
healthy 0.05
diet 0.02
…



λ=0.7

1-λ=0.3



$$p(w|\theta_1 \oplus \theta_2) = \lambda p(w|\theta_1) + (1-\lambda) p(w|\theta_2)$$

Deficiency of the Simple Mixture Model

- Adjacent words are sampled independently
- Cannot ensure passage cohesion
- Cannot capture the dependency between neighboring words

We apply the text mining algorithm to the nutrition data to find patterns …

Topic 1= text mining
$p(w|\theta_1)$

Topic 2= health
$p(w|\theta_2)$

Topic 1= text mining
$p(w|\theta_1)$

Solution=?

The Hidden Markov Model Solution

- Basic idea:
  - Make the choice of model for a word depend on the choice of model for the previous word
  - Thus we model both the choice of model ("state") and the words ("observations")

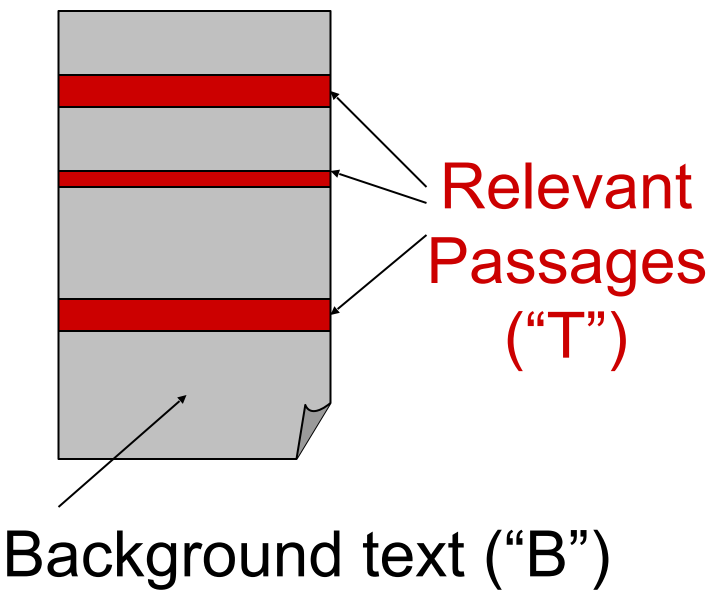
O: We apply the text mining algorithm to the nutrition data to find patterns …
S1: $\theta 1 \rightarrow \theta 1 \rightarrow \theta 1$ $\theta 1$ $\theta 1$ $\theta 1$ $\theta 1$ $\theta 1 \rightarrow \theta 1$ $\theta 1$ $\theta 1$ $\theta 1$ $\theta 1$

……

O: We apply the text mining algorithm to the nutrition data to find patterns …
S2: $\theta 1 \rightarrow \theta 1 \rightarrow \theta 1$ $\theta 1$ $\theta 1$ $\theta 1$ $\theta 1$ $\theta 1 \rightarrow \theta 2$ $\theta 1$ $\theta 1$ $\theta 1$ $\theta 1$

P(O,S1) > p(O,S2)?

Another Example: Passage Retrieval



- Strategy I: Build passage index
  - Need pre-processing
  - Query independent boundaries
  - Efficient
- Strategy II: Extract relevant passages dynamically
  - No pre-processing
  - Dynamically decide the boundaries
  - Slow

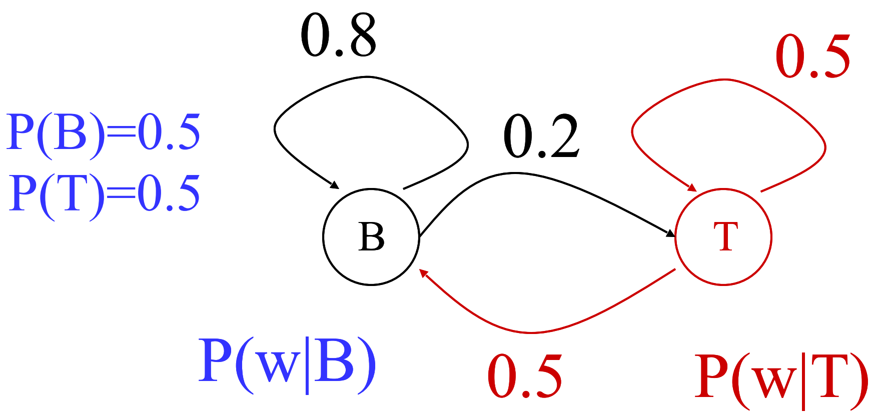
O: We apply the text mining algorithm to the nutrition data to find patterns …

S1: B T B T T T B B T T B T T

O:…......... ….. We apply the text mining … patterns. ……………….

S2: BB...BB T T T T T …. T BB….. BBBB

A Simple HMM for Recognizing Relevant Passages



the 0.2
a 0.1
we 0.01
to 0.02
…
text 0.0001
mining 0.00005
…

…
text =0.125
mining = 0.1
association =0.1
clustering = 0.05
…

Topic

Background

Parameters

Initial state prob:

p(B)= 0.5; p(T)=0.5

State transition prob:

p(B→B)=0.8 p(B→T)=0.2

p(T→B)=0.5 p(T→T)=0.5

Output prob:

P("the"|B) = 0.2 …

P("text"|T) = 0.1 …

A General Definition of HMM

$$HMM = (S, V, B, A, \Pi)$$

N states

$$S = \{s_1, \ldots, s_N\}$$

M symbols

$$V = \{v_1, \ldots, v_M\}$$

Initial state probability:

$$\Pi = \{\pi_1, \ldots, \pi_N\} \quad \sum_{i=1}^{N} \pi_i = 1$$

$$\pi_i : prob\ of\ \ starting\ \ at\ \ state\ \ s_i$$

State transition probability:

$$A = \{a_{ij}\} \quad 1 \le i, j \le N \quad \sum_{j=1}^{N} a_{ij} = 1$$

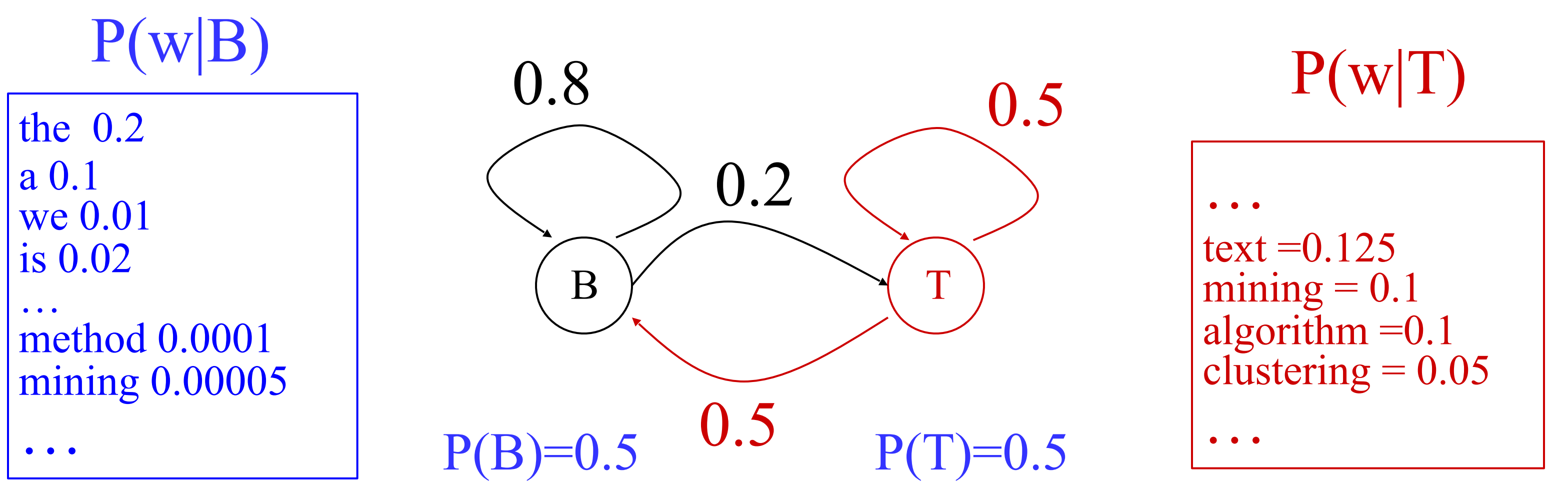
$$a_{ij} : prob\ of\ \ going\ \ s_i \rightarrow s_j$$

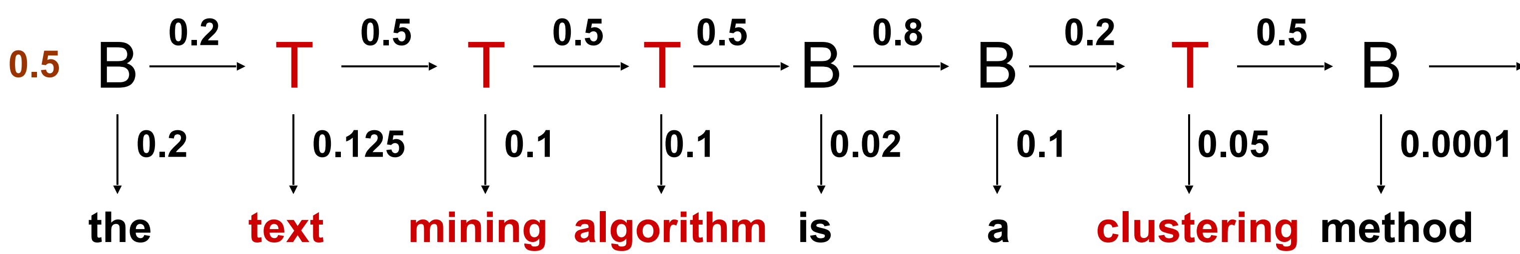
Output probability:

$$B = \{b_i(v_k)\} \quad 1 \le i \le N, 1 \le k \le M \quad \sum_{k=1}^{M} b_i(v_k) = 1$$

$$b_i(v_k) : prob\ of\ \ "generating" v_k\ at\ s_i$$

How to “Generate” Text?





P(BTT…,”the text mining…”)=p(B)p(the|B) p(T|B)p(text|T) p(T|T)p(mining|T)…

= 0.5*0.2 * 0.2*0.125 * 0.5*0.1…

HMM as a Probabilistic Model

	Time/Index:	t_1	t_2	t_3	t_4 …
Sequential data →	**Data:**	o_1	o_2	o_3	o_4 …
Random variables/ process	**Observation variable:**	O_1	O_2	O_3	O_4 …
		↑	↑	↑	↑
	Hidden state variable:	S_1 →	S_2 →	S_3 →	S_4 …

State transition prob: $p(S_1, S_2, \ldots, S_T) = p(S_1)p(S_2 \mid S_1)\ldots p(S_T \mid S_{T-1})$

Probability of observations with known state transitions:

$$p(O_1, O_2, \ldots, O_T \mid S_1, S_2, \ldots, S_T) = p(O_1 \mid S_1)p(O_2 \mid S_2)\ldots p(O_T \mid S_T)$$

Joint probability (complete likelihood): Init state distr. Output prob.

$$p(O_1, O_2, \ldots, O_T, S_1, S_2, \ldots, S_T) = p(S_1)p(O_1 \mid S_1)p(S_2 \mid S_1)p(O_2 \mid S_2)\ldots p(S_T \mid S_{T-1})p(O_T S_T)$$

Probability of observations (incomplete likelihood):

State trans. prob.

$$p(O_1, O_2, \ldots, O_T) = \sum_{S_1, \ldots S_T} p(O_1, O_2, \ldots, O_T, S_1, \ldots S_T)$$

Three Problems

1. **Decoding** – finding the most likely path

   Given: model, parameters, observations (data)

   Find: most likely states sequence (path)

$$S_1^* S_2^* \ldots S_T^* = \arg\max_{S_1 S_2 \ldots S_T} p(S_1 S_2 \ldots S_T \mid O) = \arg\max_{S_1 S_2 \ldots S_T} p(S_1 S_2 \ldots S_T, O)$$

2. **Evaluation** – computing observation likelihood

   Given: model, parameters, observations (data)

   Find: the likelihood to generate the data

$$p(O \mid \lambda) = \sum_{S_1 S_2 \ldots S_T} p(O \mid S_1 S_2 \ldots S_T) p(S_1 S_2 \ldots S_T)$$

Three Problems (cont.)

3. **Training** – estimating parameters

$$\lambda^* = \arg\max_{\lambda} \ p(O \mid \lambda)$$

- **Supervised**

  Given: model structure, labeled data(data+states sequence)

  Find: parameter values

- **Unsupervised**

  Given: model structure, data (unlabeled)

  Find: parameter values

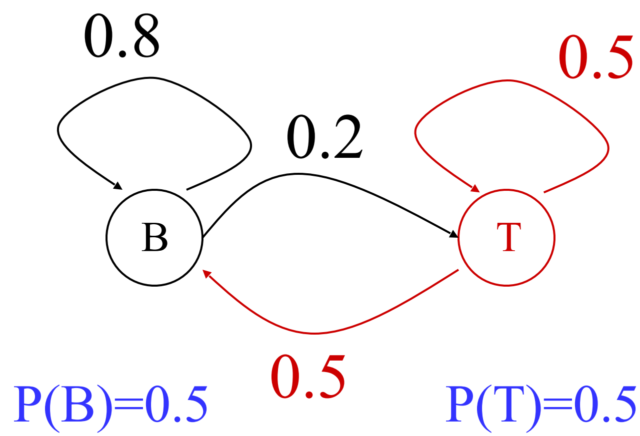
Problem I: Decoding/Parsing
Finding the most likely path

This is the most common way of using an HMM (e.g., extraction, structure analysis)…

What's the most likely path?

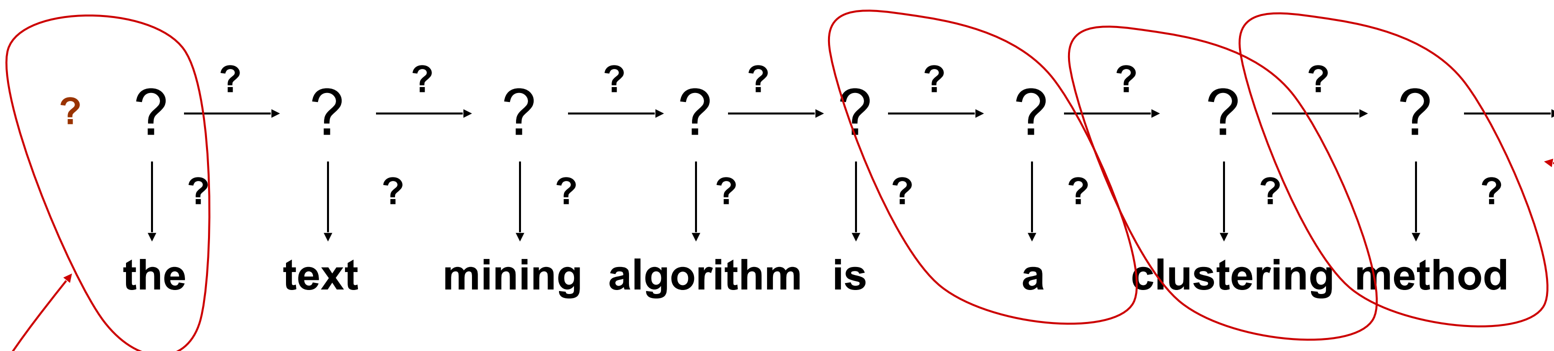
P(w|B)

the 0.2
a 0.1
we 0.01
is 0.02
…
method 0.0001
mining 0.00005
…



P(w|T)

…
text =0.125
mining = 0.1
algorithm =0.1
clustering = 0.05
…

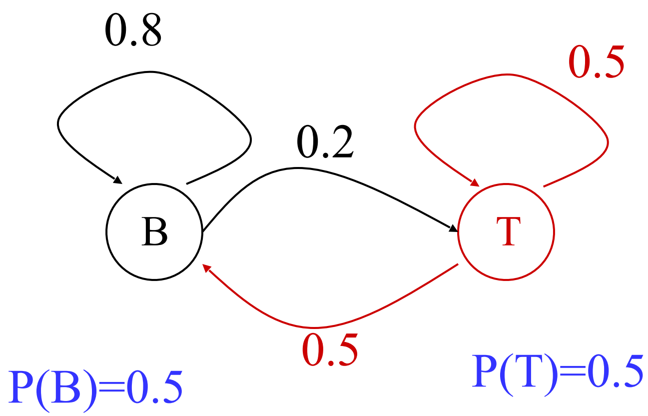


$$S_1^* S_2^* \ldots S_T^* = \arg\max_{S_1 S_2 \ldots S_T} p(S_1 S_2 \ldots S_T, O) = \arg\max_{S_1 S_2 \ldots S_T} \pi(S_1) b_{S_1}(v_{o_1}) \prod_{i=2}^{T} a_{S_{i-1} S_i} b_{S_i}(v_{o_i})$$

Viterbi Algorithm: An Example

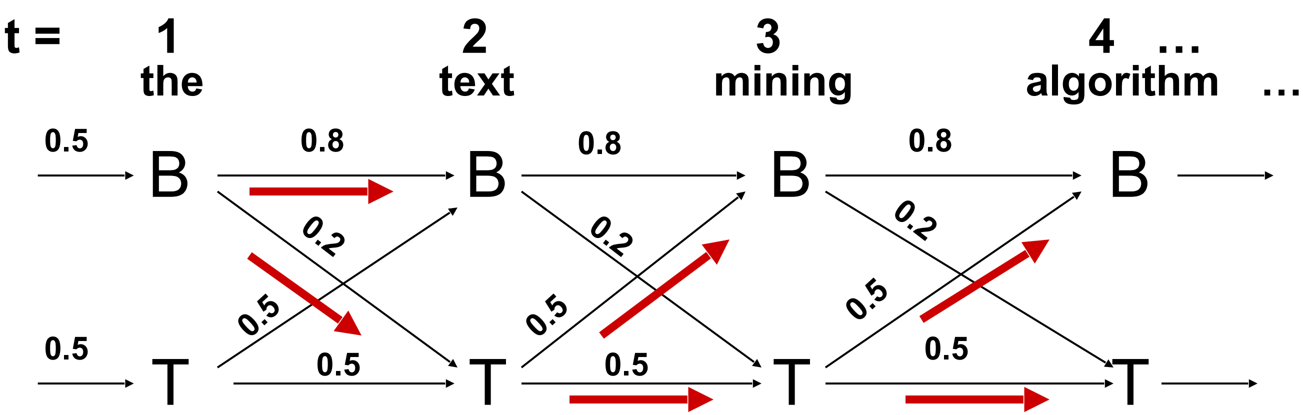
P(w|B)

the 0.2
a 0.1
we 0.01
…
algorithm 0.0001
mining 0.005
text 0.001



P(w|T)

…
the 0.05
text =0.125
mining = 0.1
algorithm =0.1
clustering = 0.05



VP(B): **0.5*0.2 (B)** **0.5*0.2*0.8*0.001(BB)** **… *0.5*0.005 (BTB)** …*0.5*0.0001(BTTB)

VP(T) **0.5*0.05(T)** **0.5*0.2*0.2*0.125(BT)** **… *0.5*0.1 (BTT)** **…*0.5*0.1(BTTT)**

Winning path!

Viterbi Algorithm

Observation: $$\max_{S_1 S_2 \ldots S_T} p(o_1 \ldots o_T, S_1 \ldots S_T) = \max_{s_i} [\max_{S_1 S_2 \ldots S_{T-1}} p(o_1 \ldots o_T, S_1 \ldots S_{T-1}, S_T = s_i)]$$

Prob. of best path ending at state i

Algorithm:
(Dynamic programming)

$$VP_t(i) = \max_{S_1 \ldots S_{t-1}} p(o_1 \ldots o_t, S_1 \ldots S_{t-1}, S_t = s_i)$$

$$q_t^*(i) = [\arg\max_{S_1 \ldots S_{t-1}} p(o_1 \ldots o_t, S_1 \ldots S_{t-1}, S_t = s_i)] \rightarrow (i)$$

Best path ending at state i

1. $VP_1(i) = \pi_i b_i(o_1), \;\; q_1^*(i) = (i), \;\; for \;\; i = 1, \ldots, N$

2. $For \;\; 1 \le t < T, \;\; VP_{t+1}(i) = \max_{1 \le j \le N} VP_t(j) a_{ji} b_i(o_{t+1}),$

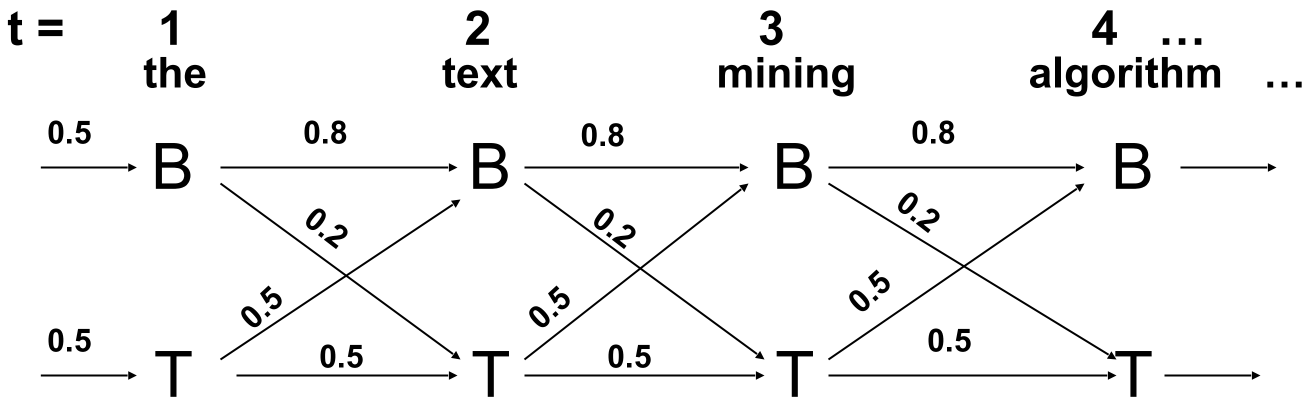
$$q_{t+1}^*(i) = q_t^*(k) \rightarrow (i), \; k = \arg\max_{1 \le j \le N} VP_t(j) a_{ji} b_i(o_{t+1}), \;\; for \;\; i = 1, \ldots, N$$

The best path is $q_T^*(i)$ Complexity: $O(TN^2)$

Problem II: Evaluation Computing the data likelihood

- Another use of an HMM, e.g., as a generative model for classification
- Also related to Problem III – parameter estimation

Data Likelihood: p(O|λ)



In general, $p(O|\lambda) = \sum_{S_1 S_2 \ldots S_T} p(O|S_1 S_2 \ldots S_T) p(S_1 S_2 \ldots S_T)$ enumerate all paths

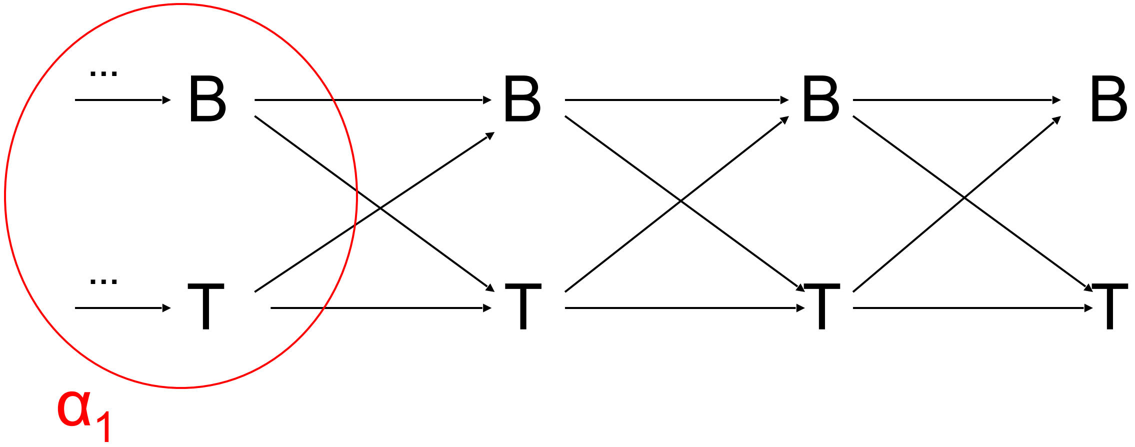
$$p(\text{"the text..."}|\lambda) = p(\text{"the text..."}|BB\ldots B)\, p(BB\ldots B)$$ BB… B

$$+\, p(\text{"the text..."}|BT\ldots B)\, p(BT\ldots B)$$ BT… B

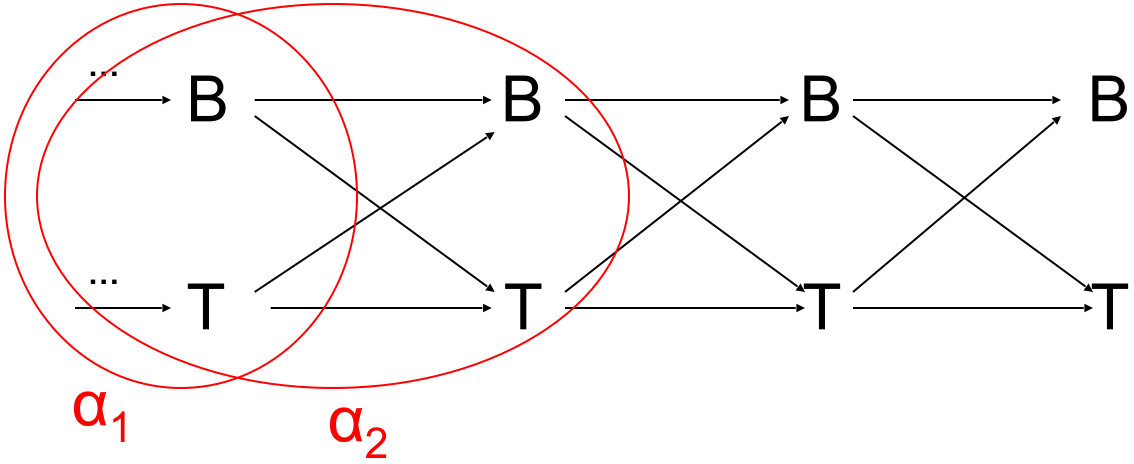
$$+ \ldots + p(\text{"the text..."}|TT\ldots T)\, p(TT\ldots T)$$ TT… T

Complexity of a naïve approach?

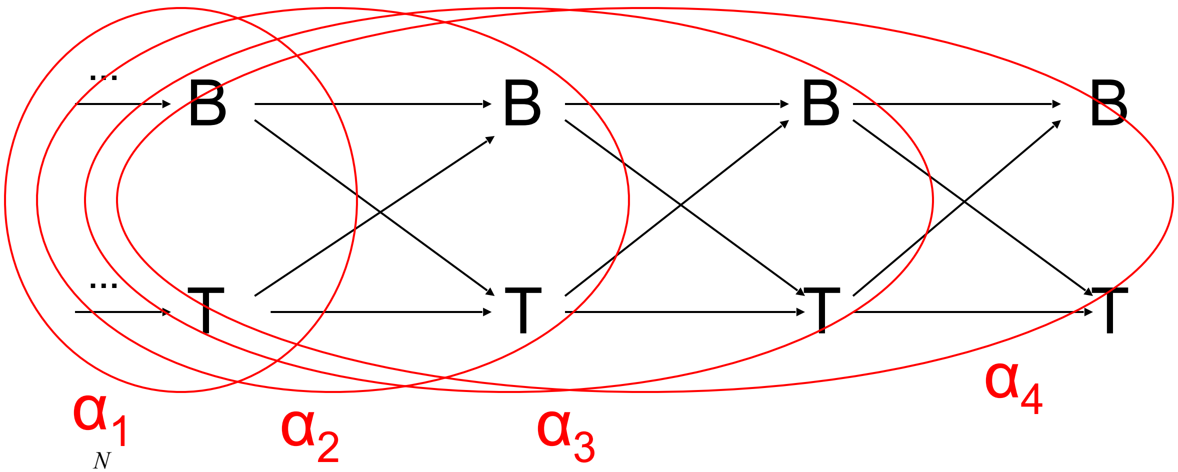
The Forward Algorithm



The Forward Algorithm



The Forward Algorithm



$$p(o_1 \ldots o_T \mid \lambda) = \sum_{i=1}^{N} \sum_{S_1 S_2 \ldots S_{T-1}} p(o_1 \ldots o_T, S_1 S_2 \ldots S_{T-1}, S_T = s_i)$$

$$\boxed{\alpha_t(i) = \sum_{S_1 S_2 \ldots S_{t-1}} p(o_1 \ldots o_t, S_1 S_2 \ldots S_{t-1}, S_t = s_i)}$$

Generating $o_1 \ldots o_t$ with ending state s_i

$$= \sum_{S_1 S_2 \ldots S_{t-1}} p(o_1 \ldots o_{t-1}, S_1 S_2 \ldots S_{t-1}) p(S_t = s_i \mid S_{t-1}) p(o_t \mid S = s_i)$$

$$= \sum_{j=1}^{N} [\sum_{S_1 S_2 \ldots S_{t-2}} p(o_1 \ldots o_{t-1}, S_1 S_2 \ldots S_{t-1} = s_j)] a_{ji} b_i(o_t)$$

$$= b_i(o_t) \sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji}$$

The data likelihood is

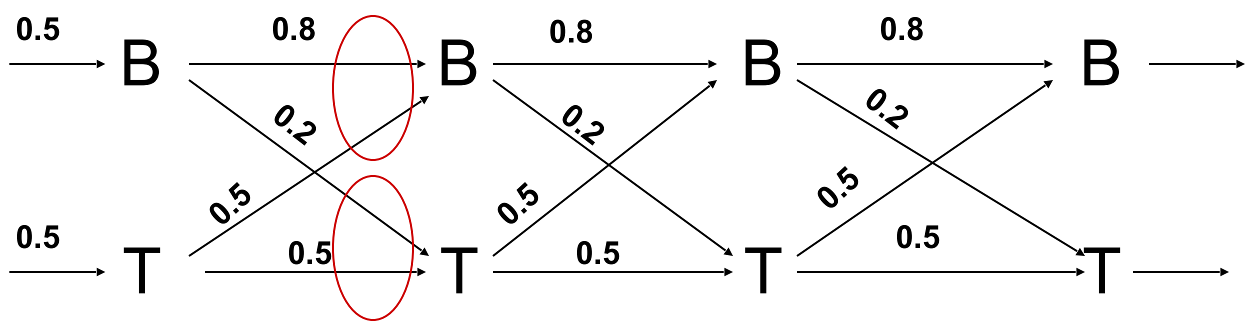
$$p(o_1 \ldots o_T \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

Complexity: $O(TN^2)$

Forward Algorithm: Example

$$p(o_1 \dots o_T \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i) \qquad \alpha_t(i) = b_i(o_t) \sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji}$$

t = 1 the 2 text 3 mining 4 algorithm … …



α_1**(B):** 0.5*p("the"|B) α_2**(B): [**α_1(B)*0.8+ α_1(T)*0.5]*p("text"|B) ……

α_1**(T):** 0.5*p("the"|T) α_2**(T): [**α_1(B)*0.2+ α_1(T)*0.5]*p("text"|T) ……

P("the text mining algorithm") = α_4(B)+ α_4(T)

The Backward Algorithm

Observation: $p(o_1 \ldots o_T \mid \lambda) = \sum_{i=1}^{N} \sum_{S_2 \ldots S_T} p(o_1 \ldots o_T, S_1 = s_i, S_2 \ldots S_T)$

$= \sum_{i=1}^{N} \pi_i b_i(o_1) \sum_{S_2 \ldots S_T} p(o_2 \ldots o_T, S_2 \ldots S_T \mid S_1 = s_i)$ **($o_1 \ldots o_t$ already generated)**

Algorithm:

$$\boxed{\beta_t(i) = \sum_{S_{t+1} \ldots S_T} p(o_{t+1} \ldots o_T, S_{t+1} \ldots S_T \mid S_t = s_i)}$$

Starting from state s_i
Generating $o_{t+1} \ldots o_T$

$$= \sum_{S_{t+1} \ldots S_T} p(o_{t+2} \ldots o_T, S_{t+2} \ldots S_T \mid S_{t+1}) p(S_{t+1} \mid S_t = s_i) p(o_{t+1} \mid S_{t+1})$$

$$= \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \sum_{S_{t+2} \ldots S_T} p(o_{t+2} \ldots o_T, S_{t+2} \ldots S_T \mid S_{t+1} = s_j)$$

$$= \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

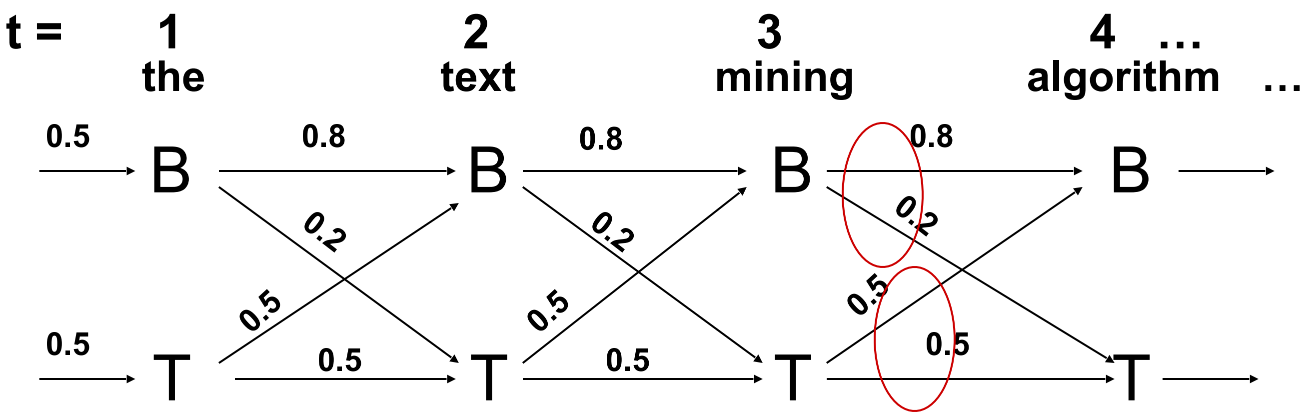
Complexity: $O(TN^2)$

The data likelihood is

$$p(o_1 \ldots o_T \mid \lambda) = \sum_{i=1}^{N} \pi_i b_i(o_1) \beta_1(i) = \sum_{i=1}^{N} \alpha_1(i) \beta_1(i) = \sum_{i=1}^{N} \alpha_t(i) \beta_t(i) \quad \textit{for any } t$$

Backward Algorithm: Example

$$p(o_1 \ldots o_T \mid \lambda) = \sum_{i=1}^{N} \pi_i b_i(o_1)\beta_1(i) = \sum_{i=1}^{N} \alpha_1(i)\beta_1(i) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) \quad \textit{for any } t$$



…

β_3**(B):** 0.8*p("alg"|B)*β_4(B)+ 0.2*p("alg"|T)*β_4(T)

β_4**(B):** 1

…

β_3**(T):** 0.5*p("alg"|B)*β_4(B)+ 0.5*p("alg"|T)*β_4(T)

β_4**(T):** 1

P("the text mining algorithm") = α_1(B)*β_1(B)+ α_1(T)*β_1(T)

= α_2(B)*β_2(B)+ α_2(T)*β_2(T)

Problem III: Training Estimating Parameters

- Where do we get the probability values for all parameters?
- Supervised vs. Unsupervised

Supervised Training

Given:

1. N – the number of states, e.g., 2, (s1 and s2)
2. V – the vocabulary, e.g., V={a,b}
3. O – observations, e.g., O=aaaaabbbbb
4. State transitions, e.g., S=1121122222

Task: Estimate the following parameters

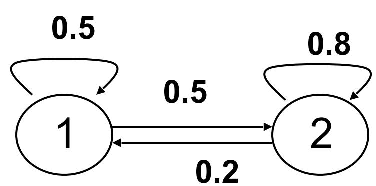
1. π_1, π_2
2. a_{11}, a_{12}, a_{22}, a_{21}
3. $b_1(a)$, $b_1(b)$, $b_2(a)$, $b_2(b)$

π_1=1/1=1; π_2=0/1=0

a_{11}=2/4=0.5; a_{12}=2/4=0.5
a_{21}=1/5=0.2; a_{22}=4/5=0.8

$b_1(a)$=4/4=1.0; $b_1(b)$=0/4=0;
b2(a)=1/6=0.167; b2(b)=5/6=0.833

P(s1)=1
P(s2)=0



P(a|s1)=1
P(b|s1)=0

P(a|s2)=0.167
P(b|s2)=0.833

Unsupervised Training

Given:

1. N – the number of states, e.g., 2, (s1 and s2)
2. V – the vocabulary, e.g., V={a,b}
3. O – observations, e.g., O=aaaaabbbbb
4. ~~State transitions, e.g., S=1121122222~~

Task: Estimate the following parameters

1. π_1, π_2
2. a_{11}, a_{12}, a_{22}, a_{21}
3. $b_1(a)$, $b_1(b)$, $b_2(a)$, $b_2(b)$

How could this be possible?

λ

Maximum Likelihood: $\lambda^* = \arg\max_\lambda p(O \mid \lambda)$

Intuition

O=aaaaabbbbb, λ

P(O,q_1|λ) **P(O,q_2|λ)** **P(O,q_K|λ)**

q_1=1111111111 **q_2=11111112211 …** **q_K=2222222222**

$$\pi_i = \frac{\sum_{k=1}^{K} p(O, q_k \mid \lambda)\delta[q_k(1) = 1]}{\sum_{k=1}^{K} p(O, q_k \mid \lambda)}$$

$$a_{ij} = \frac{\sum_{t=1}^{T-1}\sum_{k=1}^{K} p(O, q_k \mid \lambda)\delta[q_k(t) = i,\ q_k(t+1) = j]}{\sum_{t=1}^{T-1}\sum_{k=1}^{K} p(O, q_k \mid \lambda)\delta[q_k(t) = i]}$$

$$b_i(v_j) = \frac{\sum_{t=1}^{T-1}\sum_{k=1}^{K} p(O, q_k \mid \lambda)\delta[q_k(t) = i,\ o_t = v_j]}{\sum_{t=1}^{T-1}\sum_{k=1}^{K} p(O, q_k \mid \lambda)\delta[q_k(t) = i]}$$

New λ'

Computation of P(O,q_k|λ) is expensive …

Baum-Welch Algorithm

Basic "counters":

Being at state s_i at time t

$$\gamma_t(i) = p(q(t) = s_i \mid O, \lambda)$$

Being at state s_i at time t and at state s_j at time t+1

$$\xi_t(i,j) = p(q(t) = s_i, q(t+1) = s_j \mid O, \lambda)$$

Computation of counters:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}$$

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}$$

$$= \gamma_t(i) \frac{a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\beta_t(i)}$$

Complexity: $O(N^2)$

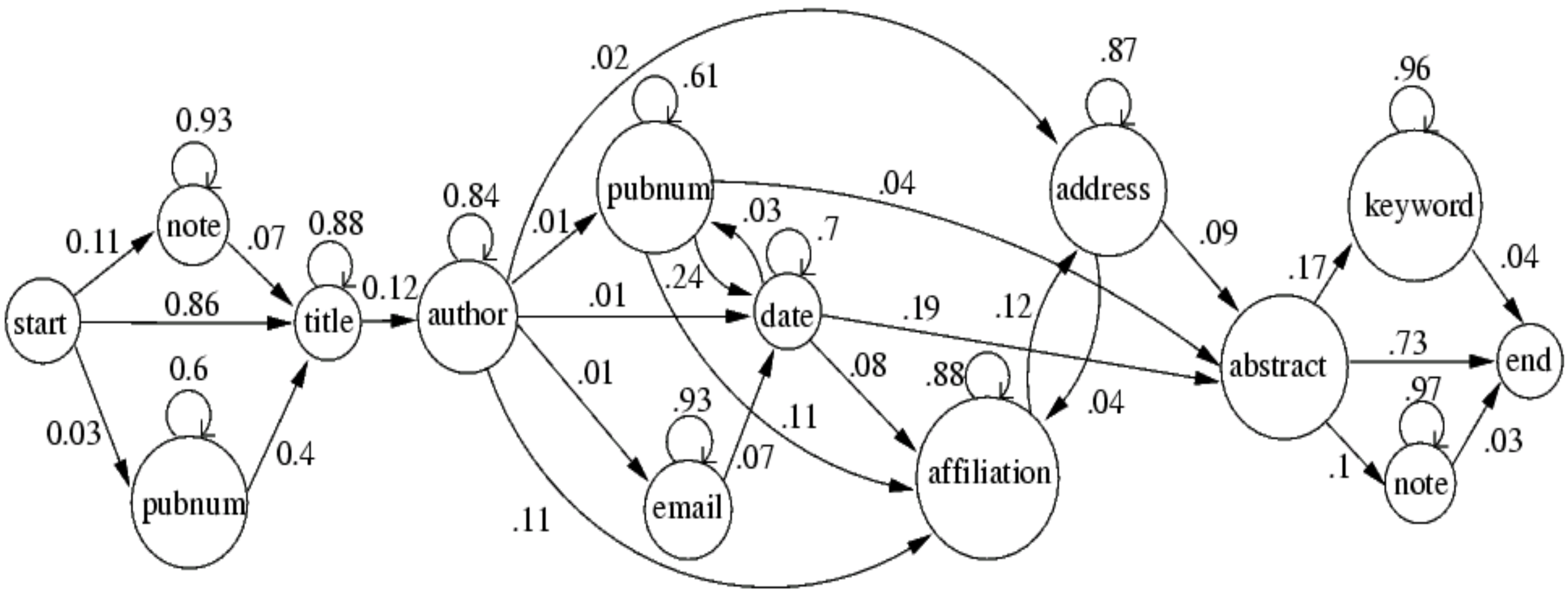
Baum-Welch Algorithm (cont.)

Updating formulas:

$$\begin{cases} \pi_i' = \gamma_1(i) \\[2ex] a_{ij}' = \dfrac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{j'=1}^{N} \sum_{t=1}^{T-1} \xi_t(i,j')} \\[4ex] b_i(v_k) = \dfrac{\sum_{t=1}^{T} \gamma_t(i)\delta[o_t = v_k]}{\sum_{t=1}^{T} \gamma_t(i)} \end{cases}$$

Overall complexity for each iteration: $O(TN^2)$

An HMM for Information Extraction (Research Paper Headers)



What You Should Know

- Definition of an HMM
- What are the three problems associated with an HMM?
- Know how the following algorithms work
  - **Viterbi algorithm**
  - **Forward & Backward algorithms**
- Know the basic idea of the Baum-Welch algorithm

Readings

- Read [Rabiner 89] sections I, II, III
- Read the “brief note”