10-701
Machine Learning

Hidden Markov models (HMMs)
What’s wrong with Bayesian networks

• Bayesian networks are very useful for modeling joint distributions
• But they have their limitations:
  - Cannot account for temporal / sequence models
  - DAG’s (no self or any other loops)

This is not a valid Bayesian network!
Hidden Markov models

- Model a set of observation with a set of hidden states
  - Robot movement
    - Observations: range sensor, visual sensor
    - Hidden states: location (on a map)
  - Speech processing
    - Observations: sound signals
    - Hidden states: parts of speech, words
  - Biology
    - Observations: DNA base pairs
    - Hidden states: Genes
Hidden Markov models

- Model a set of observation with a set of hidden states
  - Robot movement
    - **Observations:** range sensor, visual sensor
    - **Hidden states:** location (on a map)
  
1. Hidden states generate observations
2. Hidden states transition to other hidden states
Examples: Speech processing
Example: Biological data

ATGAAGCTACTGTCTTCTATCGAACAAGCATGCG
ATATTTGCCGACTTAAAAAAGCTCAAG
TGCTCCAAAGAAAAACCGAAGTGCGCCAAGTGT
CTGAAGAACAACTGAGAGTGTCGCTAC
CTGCTCCAAAACAGAGTTCTCCCGCTGACTAGG
GCACATCTGACAGAGTGGAATCAAGG
CTAGAAAGACTGGAACAGCTATTTTCTACTGATTTT
TCCTCGAGAAGACCTTGACATGATT
Contents

Preface

1 Introduction
1.1 Sequence similarity, homology, and alignment
1.2 Overview of the book
1.3 Probabilities and probabilistic models
1.4 Further reading

2 Pairwise alignment
2.1 Introduction
2.2 The scoring model
2.3 Alignment algorithms
2.4 Dynamic programming with more complex models
2.5 Heuristic alignment algorithms
2.6 Linear space alignments
2.7 Significance of scores
2.8 Deriving score parameters from alignment data
2.9 Further reading

3 Markov chains and hidden Markov models
3.1 Markov chains
3.2 Hidden Markov models
3.3 Parameter estimation for HMMs
3.4 HMM model structure
3.5 More complex Markov chains
3.6 Numerical stability of HMM algorithms
3.7 Further reading

4 Pairwise alignment using HMMs
4.1 Pair HMMs
4.2 The full probability of x and y, summing over all paths
4.3 Suboptimal alignment
4.4 The posterior probability that x_i is aligned to y_j
4.5 Pair HMMs versus FSAs for searching

page ix

1
2
2
4
10
12
12
13
17
28
32
34
36
41
45
46
48
51
62
68
72
77
79
80
81
87
89
91
95
## Contents

4.6  Further reading  

5  Profile HMMs for sequence families
   5.1  Ungapped score matrices  
   5.2  Adding insert and delete states to obtain profile HMMs  
   5.3  Deriving profile HMMs from multiple alignments  
   5.4  Searching with profile HMMs  
   5.5  Profile HMM variants for non-global alignments  
   5.6  More on estimation of probabilities  
   5.7  Optimal model construction  
   5.8  Weighting training sequences  
   5.9  Further reading  

6  Multiple sequence alignment methods
   6.1  What a multiple alignment means  
   6.2  Scoring a multiple alignment  
   6.3  Multidimensional dynamic programming  
   6.4  Progressive alignment methods  
   6.5  Multiple alignment by profile HMM training  
   6.6  Further reading  

7  Building phylogenetic trees
   7.1  The tree of life  
   7.2  Background on trees  
   7.3  Making a tree from pairwise distances  
   7.4  Parsimony  
   7.5  Assessing the trees: the bootstrap  
   7.6  Simultaneous alignment and phylogeny  
   7.7  Further reading  
   7.8  Appendix: proof of neighbour-joining theorem  

8  Probabilistic approaches to phylogeny
   8.1  Introduction  
   8.2  Probabilistic models of evolution  
   8.3  Calculating the likelihood for ungapped alignments  
   8.4  Using the likelihood for inference  
   8.5  Towards more realistic evolutionary models  
   8.6  Comparison of probabilistic and non-probabilistic methods  
   8.7  Further reading
Example: Gambling on dice outcome

- Two dices, both skewed (output model).
- Can either stay with the same dice or switch to the second dice (transition mode).
A Hidden Markov model

- A set of states \( \{s_1 \ldots s_n\} \)
  - In each time point we are in exactly one of these states denoted by \( q_t \)
- \( \Pi_i \), the probability that we start at state \( s_i \)
- A transition probability model, \( P(q_t = s_i \mid q_{t-1} = s_j) \)
- A set of possible outputs \( \Sigma \)
  - At time \( t \) we emit a symbol \( \sigma \in \Sigma \)
- An emission probability model, \( p(o_t = \sigma \mid s_i) \)
The Markov property

• A set of states \( \{s_1 \ldots s_n\} \)
  - In each time point we are in exactly one of these states denoted by \( q_t \)
• \( \Pi_i \), the probability that we start at state \( s_i \)
• A transition probability model, \( P(q_t = s_i \mid q_{t-1} = s_j) \)
  - A set of possible transitions \( \Sigma \)

An important aspect of this definition is the Markov property: \( q_{t+1} \) is conditionally independent of \( q_{t-1} \) (and any earlier time points) given \( q_t \)

More formally \( P(q_{t+1} = s_i \mid q_t = s_j) = P(q_{t+1} = s_i \mid q_t = s_j, q_{t-1} = s_j) \)
What can we ask when using a HMM?

A few examples:

• “What dice is currently being used?”
• “What is the probability of a 6 in the next role?”
• “What is the probability of 6 in any of the next 3 roles?”
Inference in HMMs

- **Q** represents a set of state: \( Q = \{s_1, s_2, \ldots, s_t\} \)
- **O** represents a set of emitted values: \( O = \{o_1, o_2, \ldots, o_t\} \)

- Computing \( P(Q) \) and \( P(q_t = s_i) \)
  - If we cannot look at observations

- Computing \( P(Q \mid O) \) and \( P(q_t = s_i \mid O) \)
  - When we have observation and care about the last state only

- Computing \( \text{argmax}_Q P(Q \mid O) \)
  - When we care about the entire path
What dice is currently being used?

- We played $t$ rounds so far
- We want to determine $P(q_t = A)$
- Let's assume for now that we cannot observe any outputs (we are blind folded)
- How can we compute this?

![Diagram](attachment:image.png)
P(q_t = A)?

• Simple answer:
Let's determine P(Q) where Q is any path that ends in A

Q = q_1, ... q_{t-1}, A

P(Q) = P(q_1, ... q_{t-1}, A) = P(A | q_1, ... q_{t-1}) P(q_1, ... q_{t-1}) =
P(A | q_{t-1}) P(q_1, ... q_{t-1}) = ... = P(A | q_{t-1}) ... P(q_2 | q_1) P(q_1)

Markov property!

Initial probability
\[ P(q_t = A) \]?

- Simple answer:
  1. Let's determine \( P(Q) \) where \( Q \) is any path that ends in \( A \)
     \[ Q = q_1, \ldots, q_{t-1}, A \]
     \[ P(Q) = P(q_1, \ldots, q_{t-1}, A) = P(A \mid q_1, \ldots, q_{t-1}) \cdot P(q_1, \ldots, q_{t-1}) = P(A \mid q_{t-1}) \cdot P(q_1, \ldots, q_{t-1}) = \cdots = P(A \mid q_{t-1}) \cdots P(q_2 \mid q_1) \cdot P(q_1) \]
  2. \( P(q_t = A) = \sum P(Q) \)

where the sum is over all sets of \( t \) states that end in \( A \)
Simple answer:

1. Let's determine $P(Q)$ where $Q$ is any path that ends in $A$

   $Q = q_1, \ldots, q_{t-1}, A$

   
   
   $P(Q) = P(q_1, \ldots, q_{t-1}, A) = P(A | q_1, \ldots, q_{t-1}) \cdot P(q_1, \ldots, q_{t-1}) = P(A | q_{t-1}) \cdot P(q_1, \ldots, q_{t-1}) = \ldots = P(A | q_1) \cdot P(q_1)$

2. $P(q_t = A) = \Sigma P(Q)$

   where the sum is over all sets of states that end in $A$

Q: How many sets $Q$ are there?

A: A lot! ($2^{t-1}$)

Not a feasible solution
P(q_t = A), the smart way

• Lets define \( p_t(i) \) as the probability of being in state \( i \) at time \( t \):
  \[ p_t(i) = p(q_t = s_i) \]
• We can determine \( p_t(i) \) by induction
  1. \( p_1(i) = \Pi_i \)
  2. \( p_t(i) = ? \)
\[ P(q_t = A), \text{ the smart way} \]

- Let's define \( p_t(i) = \text{probability state } i \text{ at time } t = p(q_t = s_i) \)
- We can determine \( p_t(i) \) by induction
  1. \( p_1(i) = \Pi_i \)
  2. \( p_t(i) = \sum_j p(q_t = s_i \mid q_{t-1} = s_j)p_{t-1}(j) \)
P(q_t = A), the smart way

• Lets define p_t(i) = probability state i at time t = p(q_t = s_i)
• We can determine p_t(i) by induction
  1. p_1(i) = Π_i
  2. p_t(i) = Σ_j p(q_t = s_i | q_{t-1} = s_j)p_{t-1}(j)

This type of computation is called dynamic programming

Complexity: O(n^2*t)

Number of states in our HMM

<table>
<thead>
<tr>
<th>Time / state</th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s2</td>
<td>.7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Inference in HMMs

• Computing $P(Q)$ and $P(q_t = s_i)$

• Computing $P(Q \mid O)$ and $P(q_t = s_i \mid O)$

• Computing $\text{argmax}_Q P(Q)$
But what if we observe outputs?

- So far, we assumed that we could not observe the outputs.
- In reality, we almost always can.

| v | P(v |A) | P(v |B) |
|---|-------|-------|
| 1 | .3    | .1    |
| 2 | .2    | .1    |
| 3 | .2    | .1    |
| 4 | .1    | .2    |
| 5 | .1    | .2    |
| 6 | .1    | .3    |
But what if we observe outputs?

- So far, we assumed that we could not observe the outputs.
- In reality, we almost always can.

\[ P(v |A) \] 
\[ P(v |B) \]

| v  | P(v |A) | P(v |B) |
|----|--------|--------|
| 1  | .3     | .1     |
| 2  | .2     | .1     |
| 3  | .2     | .1     |
| 4  | .1     | .2     |
| 5  | .1     | .2     |
| 6  | .1     | .3     |

Does observing the sequence 5, 6, 4, 5, 6, 6 change our belief about the state?
We want to compute $P(q_t = A | O_1 \ldots O_t)$

For ease of writing we will use the following notations (commonly used in the literature)

- $a_{j,i} = P(q_t = s_i | q_{t-1} = s_j)$
- $b_i(o_t) = P(o_t | s_i)$

Transition probability

Emission probability
P(q_t = A) when outputs are observed

- We want to compute P(q_t = A | O_1 ... O_t)
- Let's start with a simpler question. Given a sequence of states Q, what is P(Q | O_1 ... O_t) = P(Q | O)?
  - It is pretty simple to move from P(Q|O) to P(q_t = A | O)
  - In some cases P(Q | O) is the more important question
    - Speech processing
    - NLP
We can use Bayes rule:

\[ P(Q | O) = \frac{P(O | Q)P(Q)}{P(O)} \]

Easy, \( P(O | Q) = P(o_1 | q_1) P(o_2 | q_2) \ldots P(o_t | q_t) \)
We can use Bayes rule:

\[
P(Q | O) = \frac{P(O | Q)P(Q)}{P(O)}
\]

Easy, \( P(Q) = P(q_1) \cdot P(q_2 | q_1) \cdots P(q_t | q_{t-1}) \)
\[ P(Q | O) \]

- We can use Bayes rule:

\[ P(Q|O) = \frac{P(O|Q)P(Q)}{P(O)} \]

Hard!
What is the probability of seeing a set of observations:
- An important question in its own rights, for example classification using two HMMs

Define $\alpha_t(i) = P(o_1, o_2, \ldots, o_t \land q_t = s_i)$

$\alpha_t(i)$ is the probability that we:
1. Observe $o_1, o_2, \ldots, o_t$
2. End up at state $i$

How do we compute $\alpha_t(i)$?
Computing $\alpha_t(i)$

- $\alpha_1(i) = P(o_1 \land q_1 = i) = P(o_1 \mid q_1 = s_i) \Pi_t$

We must be at a state in time $t$

chain rule

Markov property
Example: Computing $\alpha_3(B)$

- We observed 2,3,6

$\alpha_1(A) = P(2 \land q_1 = A) = P(2 \mid q_1 = A) \Pi_A = .2 \cdot .7 = .14$, $\alpha_1(B) = .1 \cdot .3 = .03$

$\alpha_2(A) = \sum_{j=A,B} b_A(3) a_{j,A} \alpha_1(j) = .2 \cdot .8 \cdot .14 + .2 \cdot .2 \cdot .03 = 0.0236$, $\alpha_2(B) = 0.0052$

$\alpha_3(B) = \sum_{j=A,B} b_B(6) a_{j,B} \alpha_2(j) = .3 \cdot .2 \cdot .0236 + .3 \cdot .8 \cdot .0052 = 0.00264$
Where we are

- We want to compute $P(\mathbf{Q} \mid \mathbf{O})$
- For this, we only need to compute $P(\mathbf{O})$
- We know how to compute $\alpha_t(i)$

From now its easy

$$\alpha_t(i) = P(o_1, o_2 \ldots, o_t \land q_t = s_i)$$

so

$$P(\mathbf{O}) = P(o_1, o_2 \ldots, o_t) = \sum_i P(o_1, o_2 \ldots, o_t \land q_t = s_i) = \sum_i \alpha_t(i)$$

note that

$$P(\mathbf{A} \mid \mathbf{B}) = \frac{P(\mathbf{A} \land \mathbf{B})}{P(\mathbf{B})}$$
Complexity

- How long does it take to compute $P(Q \mid O)$?
- $P(Q): O(t)$
- $P(O \mid Q): O(t)$
- $P(O): O(n^2 t)$
Inference in HMMs

- Computing $P(Q)$ and $P(q_t = s_i)$
  - √
- Computing $P(Q | O)$ and $P(q_t = s_i | O)$
  - √
- Computing $\text{argmax}_Q P(Q)$
Most probable path

• We are almost done …
• One final question remains
  How do we find the most probable path, that is $Q^*$ such that

  $$P(Q^* \mid O) = \text{argmax}_Q P(Q \mid O)?$$

• This is an important path
  - The words in speech processing
  - The set of genes in the genome
  - etc.
Example

- What is the most probable set of states leading to the sequence:

  1, 2, 2, 5, 6, 5, 1, 2, 3 ?

| $v$ | $P(v | A)$ | $P(v | B)$ |
|-----|-----------|-----------|
| 1   | .3        | .1        |
| 2   | .2        | .1        |
| 3   | .2        | .1        |
| 4   | .1        | .2        |
| 5   | .1        | .2        |
| 6   | .1        | .3        |

$\Pi_A = 0.7$

$\Pi_B = 0.3$
Most probable path

$$\arg \max_Q P(Q \mid O) = \arg \max_Q \frac{P(O \mid Q)P(Q)}{P(O)}$$

$$= \arg \max_Q P(O \mid Q)P(Q)$$

We will use the following definition:

$$\delta_t(i) = \max_{q_1 \ldots q_{t-1}} p(q_1 \ldots q_{t-1} \land q_t = s_i \land O_1 \ldots O_t)$$

In other words we are interested in the most likely path from 1 to t that:

1. Ends in $S_i$
2. Produces outputs $O_1 \ldots O_t$
Computing $\delta_t(i)$

$\delta_1(i) = p(q_1 = s_i \land O_1)$

$= p(q_1 = s_i) p(O_1 \mid q_1 = s_i)$

$= \pi_i b_i(O_1)$

$\delta_t(i) = \max_{q_1 \cdots q_{t-1}} p(q_1 \cdots q_{t-1} \land q_t = s_i \land O_1 \cdots O_t)$

Q: Given $\delta_t(i)$, how can we compute $\delta_{t+1}(i)$?

A: To get from $\delta_t(i)$ to $\delta_{t+1}(i)$ we need to

1. Add an emission for time $t+1$ ($O_{t+1}$)
2. Transition to state $s_i$

$\delta_{t+1}(i) = \max_{q_1 \cdots q_t} p(q_1 \cdots q_t \land q_{t+1} = s_i \land O_1 \cdots O_{t+1})$

$= \max_j \delta_t(j) p(q_{t+1} = s_i \mid q_t = s_j) p(O_{t+1} \mid q_{t+1} = s_i)$

$= \max_j \delta_t(j) a_{j,i} b_i(O_{t+1})$
The Viterbi algorithm

\[
\delta_{t+1}(i) = \max_{q_1 \ldots q_t} p(q_1 \ldots q_t \land q_{t+1} = s_i \land O_1 \ldots O_{t+1})
\]

\[
= \max_j \delta_t(j)p(q_{t+1} = s_i \mid q_t = s_j)p(O_{t+1} \mid q_{t+1} = s_i)
\]

\[
= \max_j \delta_t(j)a_{j,i}b_i(O_{t+1})
\]

• Once again we use dynamic programming for solving \(\delta_t(i)\)

• Once we have \(\delta_t(i)\), we can solve for our \(P(Q^* \mid O)\)

By:

\[
P(Q^* \mid O) = \arg\max_Q P(Q \mid O) = \text{path defined by } \arg\max_j \delta_t(j),
\]
Inference in HMMs

- Computing $P(Q)$ and $P(q_t = s_i)$ ✓
- Computing $P(Q | O)$ and $P(q_t = s_i | O)$ ✓
- Computing $\arg\max_Q P(Q)$ ✓
What you should know

- Why HMMs? Which applications are suitable?
- Inference in HMMs
  - No observations
  - Probability of next state w. observations
  - Maximum scoring path (Viterbi)