Machine Learning for natural language processing

Hidden Markov Models

Laura Kallmeyer

Heinrich-Heine-Universität Düsseldorf

Summer 2016

Introduction

- So far, we have classified texts/observations independent from the class of the previous text/observation
- Today: sequence classifier, assigning a sequence of classes to a sequence of observations

Jurafsky & Martin (2015), chapter 8

Table of contents

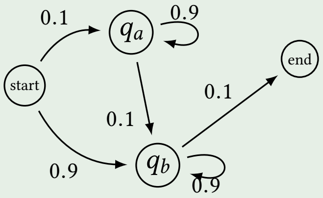
Motivation

Hidden Markov Models (**HMMs**)

- are weighted finite state automata
- with states emitting outputs.
- Transitions and emissions are weighted.

Motivation

HMM



in q_a, a is emitted with probability 0.9;
in q_a, b is emitted with probability 0.1;
in q_b, b is emitted with probability 0.9;
in q_b, a is emitted with probability 0.1;

Given a sequence bb, what is the most probable path traversed by the automaton, resulting in this output?

The states in the automaton correspond to classes, i.e., the search for the most probable path amounts to the search for the most probable class sequence.

Example

POS Tagging: the classes are POS tags, the emissions are words

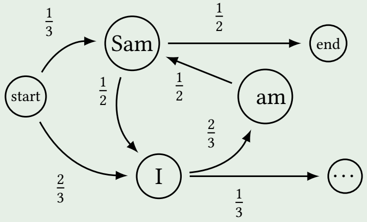
Hidden Markov Models

A **Markov chain** is a weighted automaton in which

- weights are probabilities, i.e., all weights are between 0 and 1 and the sum of the weights of all outgoing edges of a state is 1, and
- the input sequence uniquely determines the states the automaton goes through.

A Markov chain is actually a bigram language model.

Markov Chain for LM example slide 9



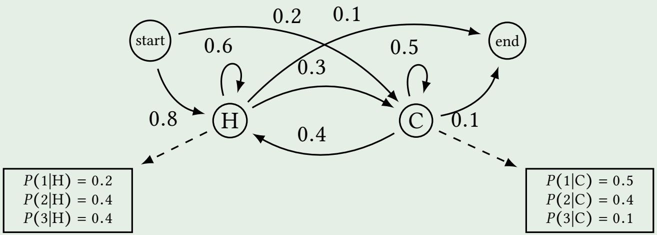
Hidden Markov Models

Markov chains are useful when we want to compute the probability for a sequence of events that we can observe.

Hidden Markov Models compute probabilities for a sequence of hidden events, given a sequence of observed events.

Example from Jurafsky & Martin (2015), after Eisner (2002)

HMM for inferring weather (hot = H, cold = C) from numbers of ice creams eaten by Jason.



Hidden Markov Models

HMM

A HMM is a tuple $\langle Q, A, O, B, q_0, q_F \rangle$, with

- $Q = \{q_1, \dots, q_N\}$ a finite set of states;
- A a $|Q| \times |Q|$ matrix, the transition probability matrix, with $0 \leq a_{ij} \leq 1$ for all $1 \leq i \leq |Q|, 1 \leq j \leq |Q|$.
- O a finite set of observations;
- a sequence B of $|Q|$ functions $b_i : O \to \mathbb{R}$, the emission probabilities with $\sum_{o \in O} b_i(o) = 1$ for all $1 \leq i \leq |Q|$
- $q_0, q_F \notin Q$ are special start and end states, associated with transition probabilities a_{0i} and a_{iF} $(0 \leq a_{0i}, a_{iF} \leq 1)$ for all $1 \leq i \leq |Q|$.

For all i, $0 \leq i \leq |Q|$, it holds that $\sum_{j=1}^{|Q|} a_{ij} + a_{iF} = 1$.

Hidden Markov Models

First order HMMs make the following simplifying assumptions:

1. **Markov assumption**: In a sequence of states $q_{i_1} \dots q_{i_n}$, we assume

$$P(q_{i_n} | q_{i_1} \dots q_{i_{n-1}}) = P(q_{i_n} | q_{i_{n-1}}) (= a_{i_{n-1} i_n})$$

2. **Output independence**: In a sequence of states $q_{i_1} \dots q_{i_n}$ with the associated output sequence $o_{i_1} \dots o_{i_n}$, we assume

$$P(o_{i_j} | q_{i_1} \dots q_{i_j} \dots q_{i_{n-1}}, o_{i_1} \dots o_{i_j} \dots o_{i_{n-1}}) = P(o_{i_j} | q_{i_j}) (= b_{i_j (o_{i_j})})$$

Hidden Markov Models

The following three fundamental problems have to be solved:

1. **Likelihood**: Given an HMM and an observation sequence, determine the likelihood of the observation sequence.
2. **Decoding**: Given an HMM and an observation sequence, discover the best hidden state sequence leading to these observations.
3. **Learning**: Given the set of states of an HMM and an observation sequence, learn the HMM parameters A and B.

Likelihood computation

Given an HMM and an observation $\mathbf{o} = \mathbf{o}_1 \dots \mathbf{o}_n$, determine the likelihood of the observation sequence $\mathbf{o}$.

For a specific state sequence $\mathbf{q} = \mathbf{q}_1 \dots \mathbf{q}_n \in Q^n$, we have

$$P(\mathbf{o}, \mathbf{q}) = P(\mathbf{o}|\mathbf{q})P(\mathbf{q})$$

and with our independence assumptions

$$P(\mathbf{o}, \mathbf{q}) = \prod_{i=1}^{n} P(\mathbf{o}_i|\mathbf{q}_i) \prod_{i=1}^{n} P(\mathbf{q}_i|\mathbf{q}_{i-1})$$

For the probability of $\mathbf{o}$, we have to sum over all possible state sequences:

$$P(\mathbf{o}) = \sum_{\mathbf{q} \in Q^n} \prod_{i=1}^{n} P(\mathbf{o}_i|\mathbf{q}_i) \prod_{i=1}^{n} P(\mathbf{q}_i|\mathbf{q}_{i-1}) P(q_F|\mathbf{q}_n)$$

Likelihood computation

Computing each element of the sum independently from the others would lead to an exponential time complexity.

Instead, we adopt the **forward algorithm** that implements some **dynamic programming**.

Idea: We fill a $n \times |Q|$ matrix α such that

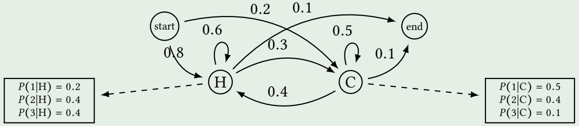
$$\alpha_{ij} = P(\mathbf{o}_1 \dots \mathbf{o}_i, \mathbf{q}_i = q_j)$$

Forward algorithm

1. $\alpha_{1j} = a_{0j} b_j(\mathbf{o}_1)$ for $1 \leq j \leq |Q|$
2. $\alpha_{ij} = \sum_{k=1}^{|Q|} \alpha_{i-1k} a_{kj} b_j(\mathbf{o}_i)$ for $1 \leq i \leq n$ and $1 \leq j \leq |Q|$
3. $P(\mathbf{o}) = \sum_{k=1}^{|Q|} \alpha_{nk} a_{kF}$

Likelihood computation

Ice cream weather example continued



$P(313)$:

H	0.32	$3.92 \cdot 10^{-2}$	$1.79 \cdot 10^{-2}$
C	$2 \cdot 10^{-2}$	$5.3 \cdot 10^{-2}$	$3.81 \cdot 10^{-3}$
	3	1	3

$$\alpha_{1j} = a_{0j} b_j(\mathbf{o}_1)$$
$$\alpha_{ij} = \sum_{k=1}^{|Q|} \alpha_{i-1k} a_{kj} b_j(\mathbf{o}_i)$$
$$P(\mathbf{o}) = \sum_{k=1}^{|Q|} \alpha_{nk} a_{kF}$$

$P(313) = 2.17 \cdot 10^{-3}$

Likelihood computation

Alternatively, we can also compute the **backward** probability β.

For a given observation $\mathbf{o}_i$ in our observation sequence, the forward probability considers the part from $\mathbf{o}_1$ to $\mathbf{o}_i$ while the backward probability considers the part from $\mathbf{o}_{i+1}$ to $\mathbf{o}_n$.

Idea: We fill a $n \times |Q|$ matrix β such that

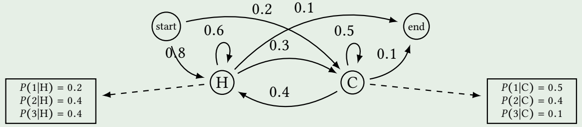
$$\beta_{ij} = P(\mathbf{o}_{i+1} \ldots \mathbf{o}_n, \mathbf{q}_i = q_j)$$

Backward algorithm

1. $\beta_{nj} = a_{jF}$ for $1 \leq j \leq |Q|$
2. $\beta_{ij} = \sum_{k=1}^{|Q|} \beta_{i+1k} a_{jk} b_k(\mathbf{o}_{i+1})$ for $1 \leq i < n$ and $1 \leq j \leq |Q|$
3. $P(\mathbf{o}) = \sum_{k=1}^{|Q|} a_{0k} b_k(\mathbf{o}_1) \beta_{1k}$

Likelihood computation

Ice cream weather example continued



observed sequence 313:

H	$6.38 \cdot 10^{-3}$	$2.7 \cdot 10^{-2}$	0.1
C	$7.4 \cdot 10^{-3}$	$2.1 \cdot 10^{-2}$	0.1
	3	1	3

$$\beta_{nj} = a_{jF}$$
$$\beta_{ij} = \sum_{k=1}^{|Q|} \beta_{i+1k} a_{jk} b_k(\mathbf{o}_{i+1})$$
$$P(\mathbf{o}) = \sum_{k=1}^{|Q|} a_{0k} b_k(\mathbf{o}_1) \beta_{1k}$$

$P(313) = 2.17 \cdot 10^{-3}$

Decoding

Given an HMM and an observation $\mathbf{o} = \mathbf{o}_1 \dots \mathbf{o}_n$, determine the most probable state sequence $\mathbf{q} = \mathbf{q}_1 \dots \mathbf{q}_n \in Q^n$ for $\mathbf{o}$.

$$\arg\max_{\mathbf{q} \in Q^n} P(\mathbf{o}, \mathbf{q}) = \arg\max_{\mathbf{q} \in Q^n} P(\mathbf{o}|\mathbf{q}) P(\mathbf{q})$$

and with our independence assumptions

$$\arg\max_{\mathbf{q} \in Q^n} P(\mathbf{o}, \mathbf{q}) = \arg\max_{\mathbf{q} \in Q^n} \prod_{i=1}^{n} P(\mathbf{o}_i|\mathbf{q}_i) \prod_{i=1}^{n} P(\mathbf{q}_i|\mathbf{q}_{i-1})$$

Decoding

For the computation, we use the **viterbi** algorithm, which is very similar to the forward algorithm except that, instead of computing sums, we compute maxs.

We fill a $n \times |Q|$ matrix v such that

$$v_{ij} = \max_{\mathbf{q} \in Q^{i-1}} P(\mathbf{o}_1 \dots \mathbf{o}_i, \mathbf{q}_1 \dots \mathbf{q}_{i-1}, \mathbf{q}_i = q_j)$$

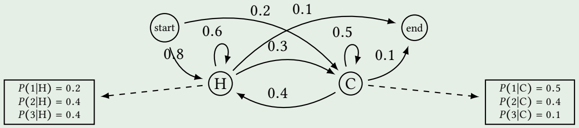
Viterbi algorithm

1. $v_{1j} = a_{0j} b_j(\mathbf{o}_1)$ for $1 \leq j \leq |Q|$
2. $v_{ij} = \max_{1 \leq k \leq |Q|} v_{i-1k} a_{kj} b_j(\mathbf{o}_i)$ for $1 \leq i \leq n$ and $1 \leq j \leq |Q|$
3. $\max_{\mathbf{q} \in Q^n} P(\mathbf{o}, \mathbf{q}) = \max_{1 \leq k \leq |Q|} v_{nk} a_{kF}$

In addition, in order to retreive the best state sequence, each entry v_{ij} is equipped with a backpointer to the state that has lead to the maximum.

Decoding

Ice cream weather example continued



output sequence 313:

H	$32 \cdot 10^{-2}$, start	$384 \cdot 10^{-4}$, H	$9216 \cdot 10^{-6}$, H
C	$2 \cdot 10^{-2}$, start	$480 \cdot 10^{-4}$, H	$24 \cdot 10^{-4}$, C
	3	1	3

$v_{1j} = a_{0j} b_j(\mathbf{o}_1)$ for $1 \leq j \leq |Q|$
$v_{ij} = \max_{1 \leq k \leq |Q|} v_{i-1k} a_{kj} b_j(\mathbf{o}_i)$ for $1 \leq i \leq n$ and $1 \leq j \leq |Q|$
$\max_{\mathbf{q} \in Q^n} P(\mathbf{o}, \mathbf{q}) = \max_{1 \leq k \leq |Q|} v_{nk} a_{kF}$

most probable weather sequence is HHH with probability $9216 \cdot 10^{-7}$

Training

Supervised HMM Training: Given a sequence of observations **o** and a corresponding sequence of states **q**, learn the parameters A and B of an HMM.
MLE via the counts of $q_i q_j$ sequences and of pairs q_i, o_j of states and observations in our training data, eventually with some smoothing.

More challenging: Unsupervised HMM Training: Given an observation sequence **o** and a state set Q, estimate A and B.

Example

- ice cream – weather case: we have a sequence of observations $\mathbf{o} \in \{1, 2, 3\}^n$ and we know that the possible states are $Q = \{H, C\}$.
- POS tagging: we have a sequence of words $\mathbf{o} \in \{w_1, w_2, \ldots, w_{|V|}\}^n$ and we know that the possible POS tags (= states) are $Q = \{NN, NNS, VBD, IN, \ldots\}$.

Task: estimate A and B of the corresponding HMMs.

Training

We use the **forward-backward** or **Baum-Welch** algorithm (Baum, 1972), a special case of the **EM** algorithm (Dempster et al., 1977).

Underlying ideas:

- We estimate parameters iteratively: we start with some parameters and use the estimated probabilities to derive better and better parameters.
- We get our estimates by computing forward probabilities and then dividing the probability mass among all the different state sequences that have contributed to a forward probability.
- Put differently, in each iteration loop, we count all possible state sequences for the given observation sequence. But, instead of counting each of them once, we use their probabilities according to the most recent parameters as counts. We perform some kind of frequency-based MLE with these **weighted** or **fractional** counts.

Training

We start with some initial A and B. In each iteration, new parameter $\hat{a}_{ij}$ and $\hat{b}_i(o)$ are computed.

Intuition:

$$\hat{a}_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i}$$

and

$$\hat{b}_i(o) = \frac{\text{expected number of times in } i \text{ observing symbol } o}{\text{expected number of times in state } i}$$

Training

More precisely, instead of the expected numbers, we use the probabilities according to the last parameters A and B and according to the counts we retreive from the observation sequence $\mathbf{o}$:

$$\hat{a}_{ij} = \frac{\text{probability of transitions from } i \text{ to } j\text{, given } \mathbf{o}}{\text{probability of transitions from state } i\text{, given } \mathbf{o}}$$

and

$$\hat{b}_i(o) = \frac{\text{probability of emitting } o \text{ in } i\text{, given } \mathbf{o}}{\text{probability of passing through state } i\text{, given } \mathbf{o}}$$

Training

For $\hat{a}_{ij}$, we first calculate the probability of being in state i at time t (observation $\mathbf{o}_t$) and in state j at time $t+1$, with the observation sequence given. This is a product of

- the probability of being in i at step t after having emitted $\mathbf{o}_1 \dots \mathbf{o}_t$, which is the forward probability α_{ti};
- the probability of the transition from i to j, which is a_{ij};
- the probability of emitting $\mathbf{o}_{t+1}$ in j, which is $b_j(\mathbf{o}_{t+1})$;
- and the probability of moving from j to q_F while emitting $\mathbf{o}_{t+2} \dots \mathbf{o}_n$, which is given by the backward probability $\beta_{t+1,j}$

divided by the probability of the observation since this is taken as given, i.e., divided by $P(\mathbf{o})$.

Training

We define $\xi_t(i,j)$ as this probability, i.e., the probability of being in i at time t, i.e., at observation $\mathbf{o}_t$ and in j when emitting $\mathbf{o}_{t+1}$, given the observation sequence $\mathbf{o}$:

$$\xi_t(i,j) = P(\mathbf{q}_t = q_i, \mathbf{q}_{t+1} = q_j | \mathbf{o})$$

According to the previous slide, this can be computed as

Transition E-step

$$\xi_t(i,j) = \frac{\alpha_{ti} a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1,j}}{P(\mathbf{o})} \text{ for } i,j \in \{1, \ldots, |Q|\}$$

This is the **E (expectation)** step for the transition probabilities.

Training

From these $\xi_t(i,j)$ values, we can estimate new transition probabilities towards maximizing the observed data:

Remember that we want to have

$$\hat{a}_{ij} = \frac{\text{probability of transitions from } i \text{ to } j\text{, given } \mathbf{o}}{\text{probability of transitions from state } i\text{, given } \mathbf{o}}$$

With the previously defined $\xi_t(i,j)$, we obtain

Transition M-step

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{n-1} \xi_t(i,j)}{\sum_{t=1}^{n-1} \sum_{k=1}^{|Q|} \xi_t(i,k) + \frac{\alpha_{n,i} a_{iF}}{P(\mathbf{o})}} \text{ for } i,j \in \{1,\dots,|Q|\}.$$

$$\hat{a}_{0i} = \frac{a_{0i} b_i(\mathbf{o}_1)\beta_{1,i}}{\sum_{k=1}^{|Q|} a_{0k} b_k(\mathbf{o}_1)\beta_{1,k}} \text{ and } \hat{a}_{iF} = \frac{\frac{\alpha_{n,i} a_{iF}}{P(\mathbf{o})}}{\sum_{t=1}^{n-1} \sum_{k=1}^{|Q|} \xi_t(i,k) + \frac{\alpha_{n,i} a_{iF}}{P(\mathbf{o})}} \text{ for } 1 \le i \le |Q|$$

This is the **M (maximization)** step for the transition probabilities.

Training

For $\hat{b}_i(o)$, we calculate the probability of being in state i at time t given the observation sequence $\mathbf{o}$, which we call $\gamma_t(i)$.

Emission E-step

$$\gamma_t(i) = P(\mathbf{q}_t = q_i|\mathbf{o}) = \frac{\alpha_{ti}\beta_{ti}}{P(\mathbf{o})}$$

This is the **E (expectation)** step for the emission probabilities.

Training

From these $\gamma_t(i)$ values, we can estimate new emission probabilities towards maximizing the observed data:

Remember that we want to have

$$\hat{b}_i(o) = \frac{\text{probability of emitting } o \text{ in } i\text{, given } \mathbf{o}}{\text{probability of passing through state } i\text{, given } \mathbf{o}}$$

Emission M-step

$$\hat{b}_i(o) = \frac{\sum_{1 \leq t \leq n, \mathbf{o}_t = o} \gamma_t(i)}{\sum_{t=1}^{n} \gamma_t(i)} \text{ for } 1 \leq i \leq |Q|$$

This is the **M (maximization)** step for the emission probabilities.

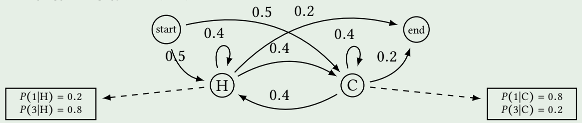
Training

```
Forward-backward algorithm (input observations $\mathbf{o} \in O^n$, state set $Q$):
**initialize** $A$ and $B$
**iterate** until convergence:
  **E-step**
    $\gamma_t(i) = \frac{\alpha_{ti}\beta_{ti}}{P(\mathbf{o})}$ for all $1 \leq t \leq n, 1 \leq i \leq |Q|$
    $\xi_t(i,j) = \frac{\alpha_{ti} a_{ij} b_j(\mathbf{o}_{t+1})\beta_{t+1,j}}{P(\mathbf{o})}$ for all $1 \leq t \leq n-1, 1 \leq i,j \leq |Q|$
  **M-step**
    $\hat{a}_{ij} = \frac{\sum_{t=1}^{n-1} \xi_t(i,j)}{\sum_{t=1}^{n-1} \sum_{k=1}^{|Q|} \xi_t(i,k) + \frac{\alpha_{n,i} a_{iF}}{P(\mathbf{o})}}$ for all $1 \leq i,j \leq |Q|$
    $\hat{a}_{0i} = \frac{a_{0i} b_i(\mathbf{o}_1)\beta_{1,i}}{\sum_{k=1}^{|Q|} a_{0k} b_k(\mathbf{o}_1)\beta_{1,k}}$ and $\hat{a}_{iF} = \frac{\frac{\alpha_{n,i} a_{iF}}{P(\mathbf{o})}}{\sum_{t=1}^{n-1} \sum_{k=1}^{|Q|} \xi_t(i,k) + \frac{\alpha_{n,i} a_{iF}}{P(\mathbf{o})}}$ for $1 \leq i \leq |Q|$
    $\hat{b}_i(o) = \frac{\sum_{1 \leq t \leq n, \mathbf{o}_t = o} \gamma_t(i)}{\sum_{t=1}^{n} \gamma_t(i)}$
**return** $A$, $B$
```

Training

(Simplified) ice cream weather example

Initialized HMM:



observed sequence 31:

$$\alpha: \begin{array}{c|cc} \text{H} & 0.4 & 4 \cdot 10^{-2} \\ \text{C} & 1 \cdot 10^{-1} & 0.16 \\ \hline t & 1 & 2 \end{array} \qquad \beta: \begin{array}{c|cc} \text{H} & 8 \cdot 10^{-2} & 0.2 \\ \text{C} & 8 \cdot 10^{-2} & 0.2 \\ \hline t & 1 & 2 \end{array} \qquad P(31) = 4 \cdot 10^{-2}$$

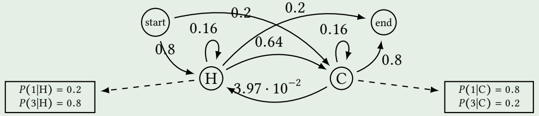
E-step: γ:

t	H	C
1	0.8	0.2
2	0.2	0.8

ξ_1:

	$j=$H	$j=$C
$i=$H	0.16	0.64
$i=$C	$3.97 \cdot 10^{-2}$	0.16

M-step:

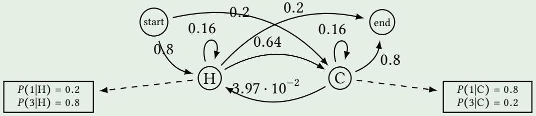


$P(31) = 0.27$

Training

(Simplified) ice cream weather example

Second step:



α

H	0.64	$2.08 \cdot 10^{-2}$
C	$4 \cdot 10^{-2}$	0.33
t	1	2

β

H	0.42	0.2
C	0.1	0.8
t	1	2

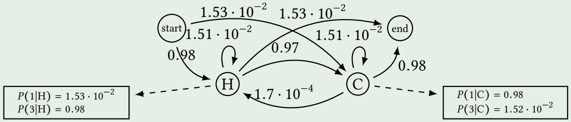
E-step: γ:

t	H	C
1	0.98	$1.53 \cdot 10^{-2}$
2	$1.53 \cdot 10^{-2}$	0.98

ξ

	j =H	j =C
i =H	$1.51 \cdot 10^{-2}$	0.97
i =C	$1.7 \cdot 10^{-4}$	$1.51 \cdot 10^{-2}$

M-step:

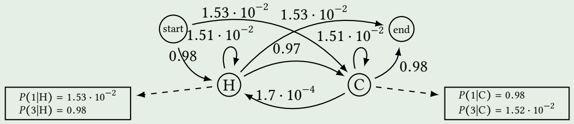


$P(31) = 0.91$

Training

(Simplified) ice cream weather example

Third step:



α

H	0.97	$2.1 \cdot 10^{-4}$
C	$2.3 \cdot 10^{-4}$	0.93
t	1	2

β

H	0.94	$1.53 \cdot 10^{-2}$
C	$1.46 \cdot 10^{-2}$	0.98
t	1	2

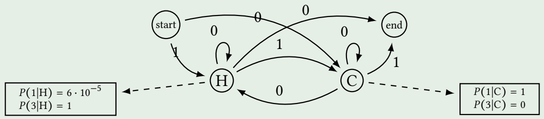
E-step: γ:

t	H	C
1	1	0
2	0	1

ξ

	j =H	j =C
i =H	0	1
i =C	0	0

M-step:



$P(31) = 1$

Training

Note that we can also leave out the factor $\frac{1}{P(\mathbf{o})}$, the result is the same:

```
**initialize** $A$ and $B$
**iterate** until convergence:
  **E-step**
    $\phi_t(i) = \alpha_{ti}\beta_{ti}$ for all $1 \leq t \leq n, 1 \leq i \leq |Q|$
    $\psi_t(i,j) = \alpha_{ti} a_{ij} b_j(\mathbf{o}_{t+1})\beta_{t+1,j}$ for all $1 \leq t \leq n-1, 1 \leq i,j \leq |Q|$
  **M-step**
    $\hat{a}_{ij} = \frac{\sum_{t=1}^{n-1} \xi_t(i,j)}{\sum_{t=1}^{n-1}\sum_{k=1}^{|Q|} \psi_t(i,k) + \alpha_{n,i} a_{iF}}$ for all $1 \leq i,j \leq |Q|$
    $\hat{a}_{0i} = \frac{a_{0i} b_i(\mathbf{o}_1)\beta_{1,i}}{\sum_{k=1}^{|Q|} a_{0k} b_k(\mathbf{o}_1)\beta_{1,k}}$ and $\hat{a}_{iF} = \frac{\alpha_{n,i} a_{iF}}{\sum_{t=1}^{n-1}\sum_{k=1}^{|Q|} \xi_t(i,k) + \alpha_{n,i} a_{iF}}$ for $1 \leq i \leq |Q|$
    $\hat{b}_i(o) = \frac{\sum_{1 \leq t \leq n, \mathbf{o}_t = o} \phi_t(i)}{\sum_{t=1}^{n} \phi_t(i)}$
**return** $A$, $B$
```

References

Baum, L. E. 1972. An inequality and ssociated maximization technique in statistical estimation for jprobabilistic functions of markov processes. In O. Shisha (ed.), *Inequalities iii: Proceedigns of the 3rd symposiom on inequalities*, 1–8. University of California, Los Angeles: Academic Press.

Dempster, A. P., N. M. Laird & D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* 39(1). 1–21.

Eisner, Jason. 2002. An interactive spreadsheet for teaching the forward-backward algorithm. In *Proceedings of the acl workshop on effective tools and methodologies for teaching nlp and cl*, 10–18.

Jurafsky, Daniel & James H. Martin. 2015. Speech and language processing. an introduction to natural language processing, computational linguistics, and speech recognition. Draft of the 3rd edition.