Hidden Markov Models

Hidden Markov Models

A Tutorial for the Course *Computational Intelligence*

`http://www.igi.tugraz.at/lehre/CI`

Barbara Resch (modified by Erhard and Car☀line Rank)
Signal Processing and Speech Communication Laboratory
Inffeldgasse 16c

Abstract

This tutorial gives a gentle introduction to Markov models and hidden Markov models (HMMs) and relates them to their use in automatic speech recognition. Additionally, the Viterbi algorithm is considered, relating the most likely state sequence of a HMM to a given sequence of observations.

Usage

To make full use of this tutorial you should

1. Download the file `HMM.zip`[1] which contains this tutorial and the accompanying Matlab programs.
2. Unzip `HMM.zip` which will generate a subdirectory named `HMM/matlab` where you can find all the Matlab programs.
3. Add the folder `HMM/matlab` and the subfolders to the Matlab search path with a command like `addpath('C:\Work\HMM\matlab')` if you are using a Windows machine or `addpath('/home/jack/HMM/matlab')` if you are using a Unix/Linux machine.

Sources

This tutorial is based on

- "Markov Models and Hidden Markov Models - A Brief Tutorial" International Computer Science Institute Technical Report TR-98-041, by Eric Fosler-Lussier,
- EPFL lab notes "Introduction to Hidden Markov Models" by Hervé Bourlard, Sacha Krstulović, and Mathew Magimai-Doss, and
- HMM-Toolbox (also included in BayesNet Toolbox) for Matlab by Kevin Murphy.

1 Markov Models

Let's talk about the weather. Let's say in Graz, there are three types of weather: sunny ☀, rainy 🌧, and foggy 🌫. Let's assume for the moment that the weather lasts all day, i.e., it doesn't change from rainy to sunny in the middle of the day.

Weather *prediction* is about trying to guess what the weather will be like tomorrow based on the observations of the weather in the past (the *history*). Let's set up a *statistical model* for weather prediction:

[1] `http://www.igi.tugraz.at/lehre/CI/tutorials/HMM.zip`

We collect statistics on what the weather q_n is like today (on day n) depending on what the weather was like yesterday q_{n-1}, the day before q_{n-2}, and so forth. We want to find the following conditional probabilities

$$P(q_n|q_{n-1}, q_{n-2}, ..., q_1), \tag{1}$$

meaning, the probability of the unknown weather at day n, $q_n \in \{\text{☀}, \text{🌧}, \text{☁}\}$, depending on the (known) weather $q_{n-1}, q_{n-2}, \ldots$ of the past days.

Using the probability in eq. 1, we can make probabilistic predictions of the type of weather for tomorrow and the next days using the observations of the weather history. For example, if we knew that the weather for the past three days was $\{\text{☀}, \text{☀}, \text{☁}\}$ in chronological order, the probability that tomorrow would be 🌧 is given by:

$$P(q_4 = \text{🌧}|q_3 = \text{☁}, q_2 = \text{☀}, q_1 = \text{☀}). \tag{2}$$

This probability could be inferred from the relative frequency (the *statistics*) of past observations of weather sequences $\{\text{☀}, \text{☀}, \text{☁}, \text{🌧}\}$.

Here's one problem: the larger n is, the more observations we must collect. Suppose that $n = 6$, then we have to collect statistics for $3^{(6-1)} = 243$ past histories. Therefore, we will make a simplifying assumption, called the Markov assumption:

For a sequence $\{q_1, q_2, ..., q_n\}$:

$$P(q_n|q_{n-1}, q_{n-2}, ..., q_1) = P(q_n|q_{n-1}). \tag{3}$$

This is called a *first-order Markov assumption*: we say that the probability of a certain observation at time n only depends on the observation q_{n-1} at time $n-1$. (A second-order Markov assumption would have the probability of an observation at time n depend on q_{n-1} and q_{n-2}. In general, when people talk about a Markov assumption, they usually mean the first-order Markov assumption.) A system for which eq. 3 is true is a *(first-order) Markov model*, and an output sequence $\{q_i\}$ of such a system is a *(first-order) Markov chain*.

We can also express the probability of a certain sequence $\{q_1, q_2, \ldots, q_n\}$ (the joint probability of certain past and current observations) using the Markov assumption:[2]

$$P(q_1, ..., q_n) = \prod_{i=1}^{n} P(q_i|q_{i-1}). \tag{4}$$

The Markov assumption has a profound affect on the number of histories that we have to find statistics for – we now only need $3 \cdot 3 = 9$ numbers ($P(q_n|q_{n-1})$ for every possible combination of $q_n, q_{n-1} \in \{\text{☀}, \text{🌧}, \text{☁}\}$) to characterize the probabilities of all possible sequences. The Markov assumption may or may not be a valid assumption depending on the situation (in the case of weather, it's probably not valid), but it is often used to simplify modeling.

So let's arbitrarily pick some numbers for $P(q_{\text{tomorrow}}|q_{\text{today}})$, as given in table 1 (note, that – whatever the weather is today – there *certainly is some kind of weather* tomorrow, so the probabilities in every *row* of table 1 sum up to one).

For first-order Markov models, we can use these probabilities to draw a probabilistic finite state automaton. For the weather domain, we would have three states, $S = \{\text{☀}, \text{🌧}, \text{☁}\}$, and every day there would be a possibility $p(q_n|q_{n-1})$ of a transition to a (possibly different) state according to the probabilities in table 1. Such an automaton would look like shown in figure 1.

1.0.1 Examples

1. Given that today the weather is ☀, what's the probability that tomorrow is ☀ and the day after is 🌧?

[2] One question that comes to mind is "What is q_0?" In general, one can think of q_0 as the *start word*, so $P(q_1|q_0)$ is the probability that q_1 can start a sentence. We can also just multiply a *prior probability* of q_1 with the product of $\prod_{i=2}^{n} P(q_i|q_{i-1})$, it's just a matter of definitions.

Today's weather	Tomorrow's weather		
	☀	🌧	🌫
☀	0.8	0.05	0.15
🌧	0.2	0.6	0.2
🌫	0.2	0.3	0.5

Table 1: Probabilities $p(q_{n+1}|q_n)$ of tomorrow's weather based on today's weather

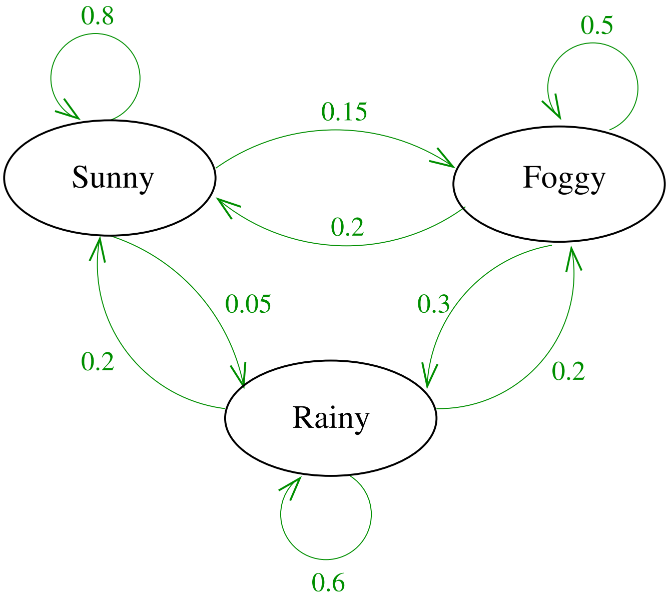


Figure 1: Markov model for the Graz weather with state transition probabilities according to table 1

Using the Markov assumption and the probabilities in table 1, this translates into:

$$\begin{aligned} P(q_2 = \text{☀}, q_3 = \text{🌧} | q_1 = \text{☀}) &= P(q_3 = \text{🌧} | q_2 = \text{☀}, q_1 = \text{☀}) \cdot P(q_2 = \text{☀} | q_1 = \text{☀}) \\ &= P(q_3 = \text{🌧} | q_2 = \text{☀}) \cdot P(q_2 = \text{☀} | q_1 = \text{☀}) \qquad \text{(Markov assumption)} \\ &= 0.05 \cdot 0.8 \\ &= 0.04 \end{aligned}$$

You can also think about this as moving through the automaton (figure 1), multiplying the probabilities along the path you go.

2. Assume, the weather yesterday was $q_1 = \text{🌧}$, and today it is $q_2 = \text{🌫}$, what is the probability that tomorrow it will be $q_3 = \text{☀}$?

$$\begin{aligned} P(q_3 = \text{☀} | q_2 = \text{🌫}, q_1 = \text{🌧}) &= P(q_3 = \text{☀} | q_2 = \text{🌫}) \qquad \text{(Markov assumption)} \\ &= 0.2. \end{aligned}$$

3. Given that the weather today is $q_1 = \text{🌫}$, what is the probability that it will be 🌧 two days from now: $q_3 = \text{🌧}$. (Hint: There are several ways to get from 🌫 today to 🌧 two days from now. You have to sum over these paths.)

2 Hidden Markov Models (HMMs)

So far we heard of the Markov assumption and Markov models. So, what is a *Hidden Markov Model*? Well, suppose you were locked in a room for several days, and you were asked about the weather outside.

The only piece of evidence you have is whether the person who comes into the room bringing your daily meal is carrying an umbrella (☂) or not (☂̸).

Let's suppose the probabilities shown in table 2: The probability that your caretaker carries an umbrella is 0.1 if the weather is sunny, 0.8 if it is actually raining, and 0.3 if it is foggy.

Weather	Probability of umbrella
Sunny	0.1
Rainy	0.8
Foggy	0.3

Table 2: Probability $P(x_i|q_i)$ of carrying an umbrella ($x_i = \text{true}$) based on the weather q_i on some day i

The equation for the weather Markov process before you were locked in the room was (eq. 4):

$$P(q_1, ..., q_n) = \prod_{i=1}^{n} P(q_i|q_{i-1}).$$

However, the actual weather is *hidden* from you. Finding the probability of a certain weather $q_i \in$ {☀, 🌧, 🌫} can only be based on the observation x_i, with $x_i =$ ☂, if your caretaker brought an umbrella on day i, and $x_i =$ ☂̸ if the caretaker did not bring an umbrella. This conditional probability $P(q_i|x_i)$ can be rewritten according to Bayes' rule:

$$P(q_i|x_i) = \frac{P(x_i|q_i)P(q_i)}{P(x_i)},$$

or, for n days, and weather sequence $Q = \{q_1, \ldots, q_n\}$, as well as 'umbrella sequence' $X = \{x_1, , \ldots, x_n\}$

$$P(q_1, \ldots, q_n|x_1, \ldots, x_n) = \frac{P(x_1, \ldots, x_n|q_1, \ldots, q_n)P(q_1, \ldots, q_n)}{P(x_1, \ldots, x_n)},$$

using the probability $P(q_1, \ldots, q_n)$ of a Markov weather sequence from above, and the probability $P(x_1, \ldots, x_n)$ of seeing a particular sequence of umbrella events (e.g., {☂, ☂̸, ☂}). The probability $P(x_1, \ldots, x_n|q_1, \ldots, q_n)$ can be estimated as $\prod_{i=1}^{n} P(x_i|q_i)$, if you assume that, for all i, the q_i, x_i are independent of all x_j and q_j, for all $j \neq i$.

We want to draw conclusions from our observations (if the persons carries an umbrella or not) about the weather outside. We can therefore omit the probability of seeing an umbrella $P(x_1, \ldots, x_n)$ as it is independent of the weather, that we like to predict. We get a measure for the probability, which is proportional to the probability, and which we will refer as the likelihood L.

$$\begin{aligned} &P(q_1, \ldots, q_n|x_1, \ldots, x_n) \propto \\ &L(q_1, \ldots, q_n|x_1, \ldots, x_n) = P(x_1, \ldots, x_n|q_1, \ldots, q_n) \cdot P(q_1, \ldots, q_n) \end{aligned} \tag{5}$$

With our (first order) Markov assumption it turns to:

$$\begin{aligned} &P(q_1, \ldots, q_n|x_1, \ldots, x_n) \propto \\ &L(q_1, \ldots, q_n|x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|q_i) \cdot \prod_{i=1}^{n} P(q_i|q_{i-1}) \end{aligned} \tag{6}$$

2.0.1 Examples

1. Suppose the day you were locked in it was sunny. The next day, the caretaker carried an umbrella into the room. You would like to know, what the weather was like on this second day.

First we calculate the likelihood for the second day to be sunny:

$$\begin{aligned} L(q_2 = \text{☀} | q_1 = \text{☀}, x_2 = \text{☂}) &= P(x_2 = \text{☂} | q_2 = \text{☀}) \cdot P(q_2 = \text{☀} | q_1 = \text{☀}) \\ &= 0.1 \cdot 0.8 = 0.08, \end{aligned}$$

then for the second day to be rainy:

$$\begin{aligned} L(q_2 = \text{🌧} | q_1 = \text{☀}, x_2 = \text{☂}) &= P(x_2 = \text{☂} | q_2 = \text{🌧}) \cdot P(q_2 = \text{🌧} | q_1 = \text{☀}) \\ &= 0.8 \cdot 0.05 = 0.04, \end{aligned}$$

and finally for the second day to be foggy:

$$\begin{aligned} L(q_2 = \text{🌫} | q_1 = \text{☀}, x_2 = \text{☂}) &= P(x_2 = \text{☂} | q_2 = \text{🌫}) \cdot P(q_2 = \text{🌫} | q_1 = \text{☀}) \\ &= 0.3 \cdot 0.15 = 0.045. \end{aligned}$$

Thus, although the caretaker did carry an umbrella, it is most likely that on the second day the weather was sunny.

2. Suppose you do not know how the weather was when your were locked in. The following three days the caretaker always comes without an umbrella. Calculate the likelihood for the weather on these three days to have been $\{q_1 = \text{☀}, q_2 = \text{🌫}, q_3 = \text{☀}\}$. As you do not know how the weather is on the first day, you assume the 3 weather situations are equi-probable on this day (cf. footnote on page 3), and the *prior probability* for sun on day one is therefore $P(q_1 = \text{☀} | q_0) = P(q_1 = \text{☀}) = 1/3$.

$$\begin{aligned} &L(q_1 = \text{☀}, q_2 = \text{🌫}, q_3 = \text{☀} | x_1 = \cancel{\text{☂}}, x_2 = \cancel{\text{☂}}, x_3 = \cancel{\text{☂}}) = \\ &\qquad P(x_1 = \cancel{\text{☂}} | q_1 = \text{☀}) \cdot P(x_2 = \cancel{\text{☂}} | q_2 = \text{🌫}) \cdot P(x_3 = \cancel{\text{☂}} | q_3 = \text{☀}) \cdot \\ &\qquad P(q_1 = \text{☀}) \cdot P(q_2 = \text{🌫} | q_1 = \text{☀}) \cdot P(q_3 = \text{☀} | q_2 = \text{🌫}) = \\ &\qquad 0.9 \cdot 0.7 \cdot 0.9 \cdot 1/3 \cdot 0.15 \cdot 0.2 = 0.0057 \end{aligned} \tag{7}$$

2.1 HMM Terminology

A HMM Model is specified by:

- The *set of states* $S = \{s_1, s_2, \dots, s_{N_s}\}$, (corresponding to the three possible weather conditions above),

and a set of parameters $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}\}$:

- The *prior probabilities* $\pi_i = P(q_1 = s_i)$ are the probabilities of s_i being the *first state* of a state sequence. Collected in a vector $\boldsymbol{\pi}$. (The prior probabilities were assumed equi-probable in the last example, $\pi_i = 1/N_s$.)
- The *transition probabilities* are the probabilities to go from state i to state j: $a_{i,j} = P(q_{n+1} = s_j | q_n = s_i)$. They are collected in the matrix $\mathbf{A}$.
- The *emission probabilities* characterize the likelihood of a certain observation x, if the model is in state s_i. Depending on the kind of observation x we have:
  - for discrete observations, $x_n \in \{v_1, \dots, v_K\}$: $b_{i,k} = P(x_n = v_k | q_n = s_i)$, the *probabilities* to observe v_k if the current state is $q_n = s_i$. The numbers $b_{i,k}$ can be collected in a matrix $\mathbf{B}$. (This would be the case for the weather model, with $K = 2$ possible observations $v_1 = \text{☂}$ and $v_2 = \cancel{\text{☂}}$.)
  - for continuous valued observations, e.g., $x_n \in \mathbb{R}^D$: A set of functions $b_i(x_n) = p(x_n | q_n = s_i)$ describing the *probability densities* (probability density functions, pdfs) over the observation space for the system being in state s_i. Collected in the vector $\mathbf{B}(x)$ of functions. Emission pdfs are often parametrized, e.g, by mixtures of Gaussians.

The operation of a HMM is characterized by

- The (hidden) *state sequence* $Q = \{q_1, q_2, \ldots, q_N\}, \quad q_n \in S$, (the sequence of the weather conditions from day 1 to N).
- The *observation sequence* $X = \{x_1, x_2, \ldots, x_N\}$.

A HMM allowing for transitions from any emitting state to any other emitting state is called an *ergodic HMM*. The other extreme, a HMM where the transitions only go from one state to itself or to a unique follower is called a *left-right HMM*.

Useful formula:

- *Probability of a state sequence*: the probability of a state sequence $Q = \{q_1, q_2, \ldots, q_N\}$ coming from a HMM with parameters $\mathbf{\Theta}$ corresponds to the product of the transition probabilities from one state to the following:

$$P(Q|\mathbf{\Theta}) = \pi_{q_1} \cdot \prod_{n=1}^{N-1} a_{q_n,q_{n+1}} = \pi_{q_1} \cdot a_{q_1,q_2} \cdot a_{q_2,q_3} \cdot \ldots \cdot a_{q_{N-1},q_N} \tag{8}$$

- *Likelihood of an observation sequence given a state sequence*, or *likelihood of an observation sequence along a single path*: given an observation sequence $X = \{x_1, x_2, \ldots, x_N\}$ and a state sequence $Q = \{q_1, q_2, \ldots, q_N\}$ (of the same length) determined from a HMM with parameters $\mathbf{\Theta}$, the likelihood of X along the path Q is equal to:

$$P(X|Q, \mathbf{\Theta}) = \prod_{n=1}^{N} P(x_n|q_n, \mathbf{\Theta}) = b_{q_1,x_1} \cdot b_{q_2,x_2} \cdot \ldots \cdot b_{q_N,x_N} \tag{9}$$

  i.e., it is the product of the emission probabilities computed along the considered path.

- *Joint likelihood of an observation sequence X and a path Q*: it is the probability that X and Q occur simultaneously, $p(X, Q|\mathbf{\Theta})$, and decomposes into a product of the two quantities defined previously:

$$P(X, Q|\mathbf{\Theta}) = P(X|Q, \mathbf{\Theta}) \cdot P(Q|\mathbf{\Theta}) \qquad \text{(Bayes)} \tag{10}$$

- *Likelihood of a sequence with respect to a HMM*: the likelihood of an observation sequence $X = \{x_1, x_2, \ldots, x_N\}$ with respect to a Hidden Markov Model with parameters $\mathbf{\Theta}$ expands as follows:

$$P(X|\mathbf{\Theta}) = \sum_{\text{all } Q} P(X, Q|\mathbf{\Theta})$$

  i.e., it is the sum of the joint likelihoods of the sequence over all possible state sequences Q allowed by the model.

2.2 Trellis Diagram

A trellis diagram can be used to visualize likelihood calculations of HMMs. Figure 2 shows such a diagram for a HMM with 3 states.

Each column in the trellis shows the possible states of the weather at a certain time n. Each state in one column is connected to each state in the adjacent columns by the transition likelihood given by the the elements $a_{i,j}$ of the transition matrix $\mathbf{A}$ (shown for state 1 at time 1 in figure 2). At the bottom is the observation sequence $X = \{x_1, \ldots, x_N\}$. $b_{i,k}$ is the likelihood of the observation $x_n = v_k$ in state $q_n = s_i$ at time n.

Figure 3 shows the trellis diagram for Example 2 in sect. 2.0.1. The likelihood of the state sequence given the observation sequence can be found by simply following the path in the trellis diagram, multiplying the observation and transition likelihoods.

$$\begin{aligned} L &= \pi_{\text{☀}} \cdot b_{\text{☀},\cancel{\text{☂}}} \cdot a_{\text{☀},\text{🌧}} \cdot b_{\text{🌧},\cancel{\text{☂}}} \cdot a_{\text{🌧},\text{☀}} \cdot b_{\text{☀},\cancel{\text{☂}}} & (11) \\ &= 1/3 \cdot 0.9 \cdot 0.15 \cdot 0.7 \cdot 0.2 \cdot 0.9 & (12) \end{aligned}$$

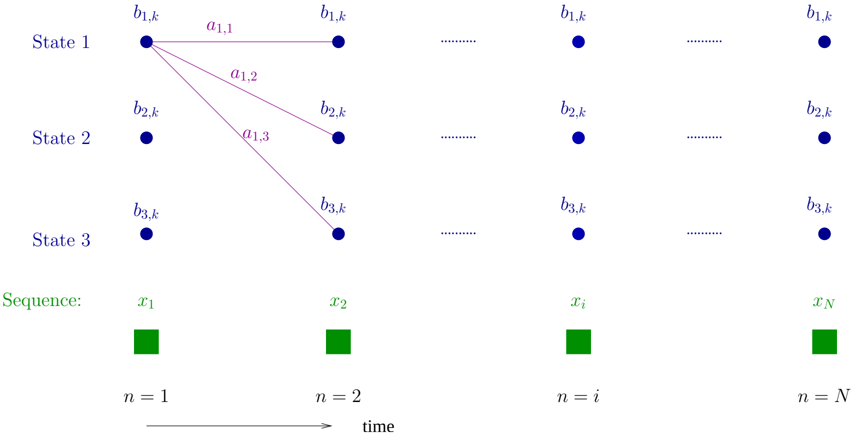


Figure 2: Trellis diagram

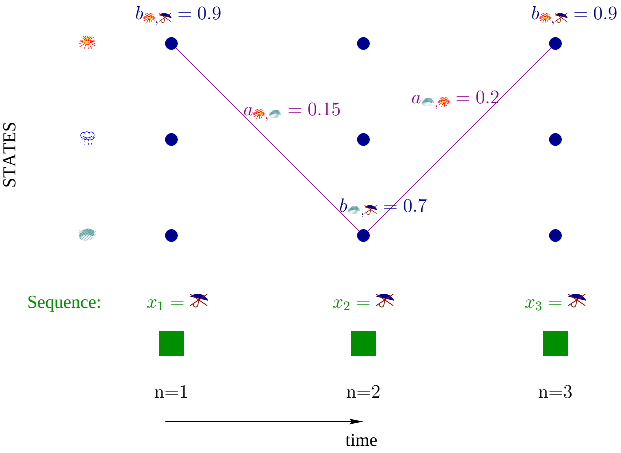


Figure 3: Trellis diagram for Example 2 in sect. 2.0.1

2.3 Generating samples from Hidden Markov Models

2.3.1 Question

Find the parameter set $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}\}$ that describes the weather HMM. Let's suppose that the prior probabilities are the same for each state.

2.3.2 Experiment

Let's generate a sample sequence X coming from our weather HMM. First you have to specify the parameter set $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}\}$ that describes the model.

```
>> %Choose the prior probabilities as you like:  E.g., first day sunny
>> Pi_w=[1 0 0]
```

The transition matrix is given in table 1:

```
>> A_w = [0.8 0.05 0.15; 0.2 0.6 0.2; 0.2 0.3 0.5]
```

As the observation probabilities in this case are discrete probabilities, they can be saved in a matrix:

```
>> B_w = [0.1 0.9; 0.8 0.2; 0.3 0.7]
```

In this observation matrix B, using the numbers from table 2, the rows correspond to the states and the columns to our 2 discrete observations, i.e., $b_{1,1} = p(\text{umbrella} = ☂)$ and $b_{1,2} = p(\text{umbrella} = ☂̸)$ is the probability of seeing an umbrella, resp. not seeing an umbrella, if the weather is in state s_1: $q_n = ☀$.

Use the function `sample_dhmm` to do several draws with the weather model. View the resulting samples and state sequences with the help of the function `plotseq_w`. In the matrix containing the data, 1 stands for 'umbrella= ☂' and 2 for 'umbrella= ☂̸'.

```
>> % drawing 1 sample of length 4 (4 days)
>> [data,hidden] = sample_dhmm(Pi_w,A_w,B_w,1,4)
>> plotseq_w(data,hidden)
```

2.3.3 Task

Write a MATLAB function `prob_path()`, that computes the joint likelihood for a given observation sequence and an assumed path:

```
>> prob = prob_path(data,path,Pi_w,A_w,B_w)
```

Input arguments are the observed sequence `data`, the assumed path `path`, and the weather HMM parameters (prior probabilities `Pi_w`, transition matrix `A_w`, and emission matrix `B_w`). Use the function `obslik = mk_dhmm_obs_lik(data,obsmat)` to produce an observation likelihood matrix where the entries correspond to $b_{i,j}$ in a trellis diagram (see figures 2 and 3). Type `help mk_dhmm_obs_lik` for more detailed information.

- Produce same random sequences, plot them and calculate their likelihood.

You can test your function to make the likelihood calculation for the 2. Example from section 2.0.1 described above.(Input parameters: `data=[2 2 2];path=[1 3 1];prior=[1/3 1/3 1/3]`.)

```
>> prob = prob_path([2 2 2],[1 3 1], [1/3 1/3 1/3], A_w, B_w)
```

2.4 HMMs for speech recognition

Let's forget about the weather and umbrellas for a moment and talk about speech. In automatic speech recognition, the task is to find the most likely sequence of words $\hat{W}$ given some acoustic input, or:

$$\hat{W} = \arg\max_{W\in\mathcal{W}} P(W|X). \tag{13}$$

Here, $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ is the sequence of "acoustic vectors" – or "feature vectors" – that are "extracted" from the speech signal, and we want to find $\hat{W}$ as the sequence of words W (out of all possible word sequences $\mathcal{W}$), that maximizes $P(W|X)$. To compare this to the weather example, the acoustic feature vectors are our observations, corresponding to the umbrella observations, and the word sequence corresponds to the weather on successive days (a day corresponds to about 10 ms), i. e., to the hidden state sequence of a HMM for speech production.

Words are made of ordered sequences of phonemes: /h/ is followed by /e/ and then by /l/ and /O/ in the word "hello". This structure can be adequately modeled by a left-right HMM, where each state corresponds to a phone. Each phoneme can be considered to produce typical feature values according to a particular probability density (possibly Gaussian) (Note, that the observed feature values $\mathbf{x}_i$ now are d-dimensional *vectors* and *continuous valued*, $\mathbf{x}_i \in \mathbb{R}^d$, and no more *discrete* values, as for the weather model where x_i could only be *true* or *false*: $x_i \in \{$☂, ~~☂~~$\}$!).

In "real world" speech recognition, the phonemes themselves are often modeled as left-right HMMs (e.g., to model separately the transition part at the begin of the phoneme, then the stationary part, and finally the transition at the end). Words are then represented by large HMMs made of concatenations of smaller phonetic HMMs.

Values used throughout the following experiments:

In the following experiments we will work with HMMs where the 'observations' are drawn from a Gaussian probability distribution. Instead of an observation matrix (as in the weather example) the observations of each state are described by the mean value and the variance of a Gaussian density.

The following 2-dimensional Gaussian pdfs will be used to model simulated observations of the vowels /a/, /i/, and /y/[3]. The observed features are the first two formants (maxima of the spectral envelope),

[3] Phonetic symbol for the usual pronunciation of 'ü'

which are characteristic for the vowel identity, e.g., for the vowel /a/ the formants typically occur around frequencies 730 Hz and 1090 Hz.

$$\text{State } s_1 \text{ (for /a/):} \quad \mathcal{N}_{/\mathrm{a}/}: \quad \boldsymbol{\mu}_{/\mathrm{a}/} = \begin{bmatrix} 730 \\ 1090 \end{bmatrix}, \quad \boldsymbol{\Sigma}_{/\mathrm{a}/} = \begin{bmatrix} 1625 & 5300 \\ 5300 & 53300 \end{bmatrix}$$

$$\text{State } s_2 \text{ (for /i/):} \quad \mathcal{N}_{/\mathrm{i}/}: \quad \boldsymbol{\mu}_{/\mathrm{i}/} = \begin{bmatrix} 270 \\ 2290 \end{bmatrix}, \quad \boldsymbol{\Sigma}_{/\mathrm{i}/} = \begin{bmatrix} 2525 & 1200 \\ 1200 & 36125 \end{bmatrix}$$

$$\text{State } s_3 \text{ (for /y/):} \quad \mathcal{N}_{/\mathrm{y}/}: \quad \boldsymbol{\mu}_{/\mathrm{y}/} = \begin{bmatrix} 440 \\ 1020 \end{bmatrix}, \quad \boldsymbol{\Sigma}_{/\mathrm{y}/} = \begin{bmatrix} 8000 & 8400 \\ 8400 & 18500 \end{bmatrix}$$

(Those densities have been used in the previous lab session.) The Gaussian pdfs now take the role of the emission matrix B in the hidden Markov models for recognition of the three vowels /a/, /i/, and /y/.

The parameters of the densities and of the Markov models are stored in the file `data.mat` (Use: `load data`). A Markov model named, e.g., `hmm1` is stored as an object with fields `hmm1.means`, `hmm1.vars`, `hmm1.trans`,`hmm1.pi`. The `means` fields contains a matrix of mean vectors, where each column of the matrix corresponds to one state s_i of the HMM (e.g., to access the means of the second state of `hmm1` use: `hmm1.means(:,2)`; the `vars` field contains a 3 dimensional array of variance matrices, where the third dimension corresponds to the state (e.g to access $\boldsymbol{\Sigma}_{/\mathrm{a}/}$ for state 1 use `hmm1.vars(:,:,1)`; the `trans` field contains the transition matrix, and the `pi` field the prior probabilities.

2.4.1 Task

Load the HMMs and make a sketch of each of the models with the transition probabilities.

2.4.2 Experiment

Generate samples using the HMMs and plot them with plotseq and plotseq2.
Use the functions `plotseq` and `plotseq2` to plot the obtained 2-dimensional data. In the resulting views, the obtained sequences are represented by a yellow line where each point is overlaid with a colored dot. The different colors indicate the state from which any particular point has been drawn.

```
>> %Example:generate sample from HMM1 of length N
>> [X,ST]=sample_ghmm(hmm1,N)
>> plotseq(X,ST) % View of both dimensions as separate sequences
>> plotseq2(X,ST,hmm1) %2D view with location of gaussian states
```

Draw several samples with the same parameters and compare. Compare the Matlab figures with your sketch of the model.
What is the effect of the different transition matrices of the HMMs on the sequences obtained during the current experiment? Hence, what is the role of the transition probabilities in the HMM?

3 Pattern Recognition with HMM

In equation 10 we expressed the joint likelihood $p(X,Q|\boldsymbol{\Theta})$ of an observation sequence X and a path Q given a model with parameters $\boldsymbol{\Theta}$.

The *likelihood of a sequence with respect to a HMM* (the likelihood of an observation sequence $X = \{x_1, x_2, \cdots, x_N\}$ for a given hidden Markov model with parameters $\boldsymbol{\Theta}$) expands as follows:

$$p(X|\boldsymbol{\Theta}) = \sum_{\text{every possible } Q} p(X,Q|\boldsymbol{\Theta}), \tag{14}$$

i.e., it is the sum of the joint likelihoods of the sequence *over all possible state sequences* allowed by the model (see Trellis diagram in figure 3).

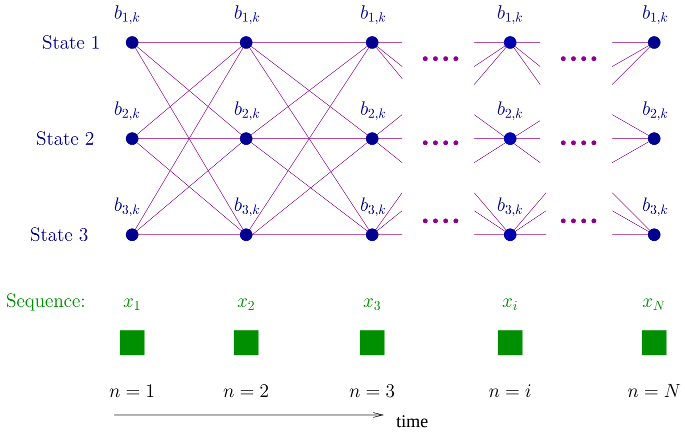


Figure 4:

Calculating the likelihood in this manner is computationally expensive, particularly for large models or long sequences. It can be done with a recursive algorithm (forward-backward algorithm), which reduces the complexity of the problem. For more information about this algorithm see [1]. It is very common using log-likelihoods and log-probabilities, computing $\log p(X|\mathbf{\Theta})$ instead of $p(X|\mathbf{\Theta})$.

3.0.1 Experiment

Classify the sequences X_1, X_2, X_3, X_4, given in the file `Xdata.mat`, in a maximum likelihood sense with respect to the four Markov models. Use the function `loglik_ghmm` to compute the likelihood of a sequence with respect to a HMM. Store the results in a matrix (they will be used in the next section).

```
>> load Xdata
>> % Example:
>> logProb(1,1) = loglik_ghmm(X1,hmm1)
>> logProb(1,2) = loglik_ghmm(X1,hmm2)
```

etc.

```
>> logProb(3,2) = loglik_ghmm(X3,hmm2)
```

etc.

Instead of typing these commands for every combination of the four sequences and models, filling the `logProb` matrix can be done automatically with the help of loops, using a command string composed of fixed strings and strings containing the number of sequence/model:

```
>> for i=1:4,
     for j=1:4,
      stri = num2str(i);
      strj = num2str(j);
      cmd = ['logProb(',stri,',',strj,') = loglik_ghmm(X',stri,',hmm',strj,');']
      eval(cmd);
    end;
  end;
```

You can find the maximum value of each row i of the matrix, giving the index of the most likely model for sequence X_i, with the MATLAB function `max`:

```
for i=1:4;
    [v,index]=max(logProb(i,:));
    disp (['X',num2str(i),' -> HMM',num2str(index)]);
end
```

i	Sequence	$\log p(X_i\|\mathbf{\Theta}_1)$	$\log p(X_i\|\mathbf{\Theta}_2)$	$\log p(X_i\|\mathbf{\Theta}_3)$	$\log p(X_i\|\mathbf{\Theta}_4)$	Most likely model
1	X_1					
2	X_2					
3	X_3					
4	X_4					

4 Optimal state sequence

In speech recognition and several other pattern recognition applications, it is useful to associate an "optimal" sequence of states to a sequence of observations, given the parameters of a model. For instance, in the case of speech recognition, knowing which frames of features "belong" to which state allows to locate the word boundaries across time. This is called *alignment* of acoustic feature sequences.

A "reasonable" optimality criterion consists of choosing the state sequence (or *path*) that has the maximum likelihood with respect to a given model. This sequence can be determined recursively via the *Viterbi algorithm*.

This algorithm makes use of two variables:

- $\delta_n(i)$ is the *highest* likelihood of a *single* path among all the paths ending in state s_i at time n:

$$\delta_n(i) = \max_{q_1,q_2,\cdots,q_{n-1}} p(q_1, q_2, \ldots, q_{n-1}, q_n = s_i, x_1, x_2, \ldots x_n|\mathbf{\Theta}) \tag{15}$$

- a variable $\psi_n(i)$ which allows to keep track of the "best path" ending in state s_i at time n:

$$\psi_n(i) = \arg\max_{q_1,q_2,\cdots,q_{n-1}} p(q_1, q_2, \ldots, q_{n-1}, q_n = s_i, x_1, x_2, \ldots x_n|\mathbf{\Theta}) \tag{16}$$

The idea of the Viterbi algorithm is to find the most probable path *for each intermediate and finally for the terminating state* in the trellis. At each time n only the most likely path leading to each state s_i 'survives'.

The Viterbi Algorithm

for a HMM with N_s states.

1. **Initialization**

$$\begin{aligned} &\delta_1(i) = \pi_i \cdot b_{i,x_1}, \quad i = 1, \ldots, N_s \\ &\psi_1(i) = 0 \end{aligned} \tag{17}$$

where π_i is the prior probability of being in state s_i at time $n = 1$.

2. **Recursion**

$$\begin{aligned} \delta_n(j) &= \max_{1\leq i\leq N_s} (\delta_{n-1}(i) \cdot a_{ij}) \cdot b_{j,x_n}, \quad \begin{matrix} 2 \leq n \leq N \\ 1 \leq j \leq N_s \end{matrix} \\ \psi_n(j) &= \arg\max_{1\leq i\leq N_s} (\delta_{n-1}(i) \cdot a_{ij}), \quad \begin{matrix} 2 \leq n \leq N \\ 1 \leq j \leq N_s \end{matrix} \end{aligned} \tag{18}$$

"Optimal policy is composed of optimal sub-policies": find the path that leads to a maximum likelihood considering the best likelihood at the previous step and the transitions from it; then multiply by the current likelihood given the current state. Hence, the best path is found by induction.

3. **Termination**

$$\begin{aligned} p^*(X|\mathbf{\Theta}) &= \max_{1\le i\le N_s} \delta_N(i) \\ q_N^* &= \arg\max_{1\le i\le N_s} \delta_N(i) \end{aligned} \tag{19}$$

   Find the best likelihood when the end of the observation sequence $t = T$ is reached.

4. **Backtracking**

$$Q^* = \{q_1^*, \ldots, q_N^*\} \quad \text{so that} \quad q_n^* = \psi_{n+1}(q_{n+1}^*), \quad n = N-1, N-2, \ldots, 1 \tag{20}$$

   Read (decode) the best sequence of states from the ψ_n vectors.

4.0.1 Example

Let's get back to our weather HMM. You don't know how the weather was when your were locked in. On the first 3 days your umbrella observations are: {no umbrella, umbrella, umbrella} ($\{\cancel{\text{☂}}, \text{☂}, \text{☂}\}$). Find the most probable weather-sequence using the Viterbi algorithm (assume the 3 weather situations to be equi-probable on day 1).

1. Initialization
   $n = 1$:

$$\begin{aligned} \delta_1(\text{☀}) &= \pi_{\text{☀}} \cdot b_{\text{☀},\cancel{\text{☂}}} = 1/3 \cdot 0.9 = 0.3 \\ \psi_1(\text{☀}) &= 0 \\ \delta_1(\text{🌧}) &= \pi_{\text{🌧}} \cdot b_{\text{🌧},\cancel{\text{☂}}} = 1/3 \cdot 0.2 = 0.0667 \\ \psi_1(\text{🌧}) &= 0 \\ \delta_1(\text{☁}) &= \pi_{\text{☁}} \cdot b_{\text{☁},\cancel{\text{☂}}} = 1/3 \cdot 0.7 = 0.233 \\ \psi_1(\text{☁}) &= 0 \end{aligned}$$

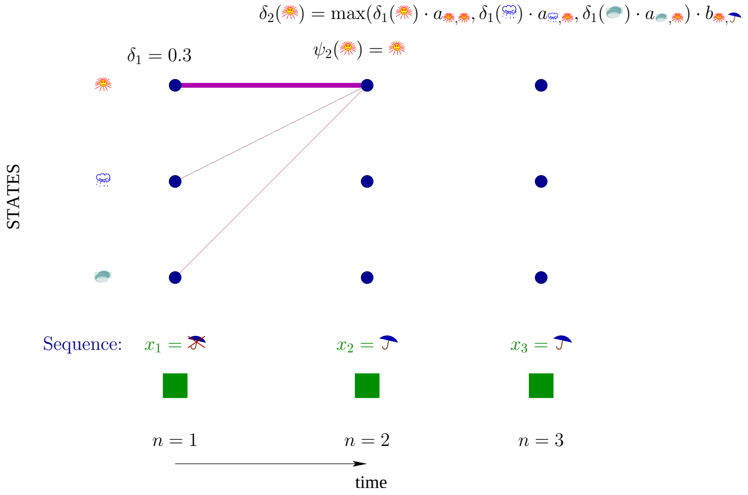
2. Recursion
   $n = 2$:
   We calculate the likelihood of getting to state '☀' from all possible 3 predecessor states, and choose the most likely one to go on with:

$$\begin{aligned} \delta_2(\text{☀}) &= \max(\delta_1(\text{☀}) \cdot a_{\text{☀},\text{☀}}, \delta_1(\text{🌧}) \cdot a_{\text{🌧},\text{☀}}, \delta_1(\text{☁}) \cdot a_{\text{☁},\text{☀}}) \ \cdot b_{\text{☀},\text{☂}} \\ &= \max(0.3 \cdot 0.8, 0.0667 \cdot 0.2, 0.233 \cdot 0.2) \ \cdot 0.1 = 0.024 \\ \psi_2(\text{☀}) &= \text{☀} \end{aligned}$$

   The likelihood is stored in δ, the most likely predecessor in ψ. See figure 5.

   The same procedure is executed with states 🌧and ☁:

$$\begin{aligned} \delta_2(\text{🌧}) &= \max(\delta_1(\text{☀}) \cdot a_{\text{☀},\text{🌧}}, \delta_1(\text{🌧}) \cdot a_{\text{🌧},\text{🌧}}, \delta_1(\text{☁}) \cdot a_{\text{☁},\text{🌧}}) \ \cdot b_{\text{🌧},\text{☂}} \\ &= \max(0.3 \cdot 0.05, 0.0667 \cdot 0.6, 0.233 \cdot 0.3) \ \cdot 0.8 = 0.056 \\ \psi_2(\text{🌧}) &= \text{☁} \\ \delta_2(\text{☁}) &= \max(\delta_1(\text{☀}) \cdot a_{\text{☀},\text{☁}}, \delta_1(\text{🌧}) \cdot a_{\text{🌧},\text{☁}}, \delta_1(\text{☁}) \cdot a_{\text{☁},\text{☁}}) \ \cdot b_{\text{☁},\text{☂}} \\ &= \max(0.3 \cdot 0.15, 0.0667 \cdot 0.2, 0.233 \cdot 0.5) \ \cdot 0.3 = 0.0350 \\ \psi_2(\text{☁}) &= \text{☁} \end{aligned}$$



$\delta_2(☀) = \max(\delta_1(☀)\cdot a_{☀,☀}, \delta_1(🌧)\cdot a_{🌧,☀}, \delta_1(🌫)\cdot a_{🌫,☀})\cdot b_{☀,☂}$
$\delta_1 = 0.3$
$\psi_2(☀) = ☀$
STATES
Sequence:
$x_1 = \cancel{☂}$
$x_2 = ☂$
$x_3 = ☂$
$n = 1$
$n = 2$
$n = 3$
time

Figure 5: Viterbi algorithm to find the most likely weather sequence. Finding the most likely path to state 'sunny' at $n = 2$.

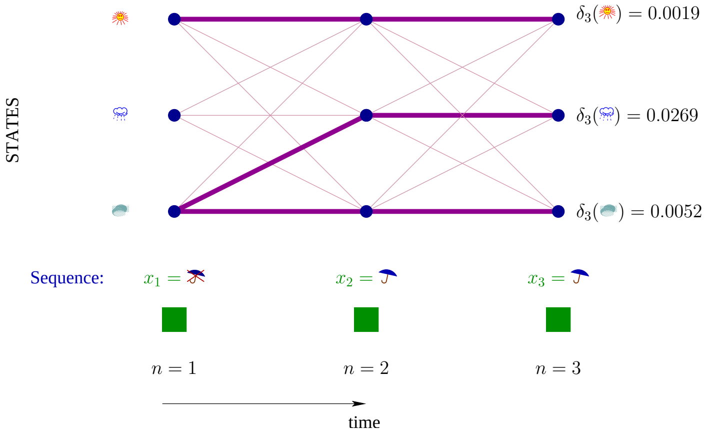


Figure 6: Viterbi algorithm to find most likely weather sequence at $n = 3$.

$n = 3:$

$$
\begin{aligned}
\delta_3(☀) &= \max(\delta_2(☀)\cdot a_{☀,☀}, \delta_2(🌧)\cdot a_{🌧,☀}, \delta_2(🌫)\cdot a_{🌫,☀})\ \cdot b_{☀,☂}\\
&= \max(0.024\cdot 0.8, 0.056\cdot 0.2, 0.035\cdot 0.2)\ \cdot 0.1 = 0.0019\\
\psi_3(☀) &= ☀\\
\delta_3(🌧) &= \max(\delta_2(☀)\cdot a_{☀,🌧}, \delta_2(🌧)\cdot a_{🌧,🌧}, \delta_2(🌫)\cdot a_{🌫,🌧})\ \cdot b_{🌧,☂}\\
&= \max(0.024\cdot 0.05, 0.056\cdot 0.6, 0.035\cdot 0.3)\ \cdot 0.8 = 0.0269\\
\psi_3(🌧) &= 🌧\\
\delta_3(🌫) &= \max(\delta_2(☀)\cdot a_{☀,🌫}, \delta_2(🌧)\cdot a_{🌧,🌫}, \delta_2(🌫)\cdot a_{🌫,🌫})\ \cdot b_{🌫,☂}\\
&= \max(0.0024\cdot 0.15, 0.056\cdot 0.2, 0.035\cdot 0.5)\ \cdot 0.3 = 0.0052\\
\psi_3(🌧) &= 🌫
\end{aligned}
$$

Finally, we get one most likely path ending in each state of the model. See figure 6.

3. Termination
The globally most likely path is determined, starting by looking for the last state of the most likely sequence.

$$P^*(X|\boldsymbol{\Theta}) = \max(\delta_3(i)) = \delta_3(🌧) = 0.0269$$
$$q_3^* = \arg\max(\delta_3(i)) = 🌧$$

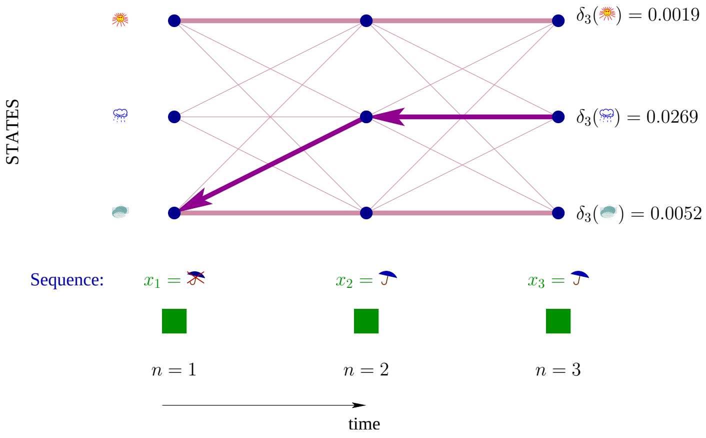


Figure 7: Viterbi algorithm to find most likely weather sequence. Backtracking.

4. Backtracking
The best sequence of states can be read from the ψ vectors. See figure 7.
$n = N - 1 = 2$:
$$q_2^* = \psi_3(q_3^*) = \psi_3(🌧) = 🌧$$
$n = N - 1 = 1$:
$$q_1^* = \psi_2(q_2^*) = \psi_2(🌧) = ☁$$
Thus the most likely weather sequence is: $Q^* = \{q_1^*, q_2^*, q_3^*\} = \{$☁, 🌧, 🌧$\}$.

4.0.2 Task

1. Write a MATLAB function `[loglik,path] = vit_ghmm(data,HMM)` to implement the Viterbi algorithm for HMMs with Gaussian emissions. Use the function `mk_ghmm_obs_lik` to calculate the observation likelihood matrix for a given sequence. Store the δ and ψ vectors in a matrix in the format of the observation likelihood matrix. You can either write the function with the algorithm exactly as given before, or switch to make the calculations in the log likelihood domain, where the multiplications of the parameters δ and ψ transform to additions. What are the advantages or disadvantages of the second method? (Think about how to implement it on a real system with limited computational accuracy, and about a HMM with a large number of states where the probabilities $a_{i,j}$ and $b_{i,k}$ might be small.)

2. Use the function `vit_ghmm` to determine the most likely path for the sequences X_1, X_2, X_3 and X_4. Compare with the state sequence $ST1, \ldots, ST4$ originally used to generate $X_1, \ldots, X_4$. (Use the function `compseq`, which provides a view of the first dimension of the observations as a time series, and allows to compare the original alignment to the Viterbi solution).
```
>> [loglik,path] = vit_ghmm(X1,hmm1); compseq(X1,ST1,path);
>> [loglik,path] = vit_ghmm(X2,hmm1); compseq(X2,ST2,path);
```
Repeat for the remaining sequences.

3. Use the function `vit_ghmm` to compute the probabilities of the sequences $X_1, \ldots, X_4$ along the best paths with respect to each model $\mathbf{\Theta}_1, \ldots, \mathbf{\Theta}_4$. Note down your results below. Compare with the log-likelihoods obtained in section 3.0.1 using the forward procedure.
```
>> diffL=logProb-logProbViterbi
```

Likelihoods along the best path:

i	Sequence	$\log p^*(X_i \mid \mathbf{\Theta}_1)$	$\log p^*(X_i \mid \mathbf{\Theta}_2)$	$\log p^*(X_i \mid \mathbf{\Theta}_3)$	$\log p^*(X_i \mid \mathbf{\Theta}_4)$	Most likely model
1	X_1					
2	X_2					
3	X_3					
4	X_4					

Difference between log-likelihoods and likelihoods along the best path:

Sequence	HMM1	HMM2	HMM3	HMM4
X_1				
X_2				
X_3				
X_4				

Question:

Is the likelihood along the best path a good approximation of the real likelihood of a sequence given a model?

References

[1] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77 (2), 257–286.