Viterbi Algorithm to Predict Aerodynamic Control Laws of an Aircraft

Objective of the experiment: To build a computational stochastic model based on Markov chains to predict the most likely sequence of events (Markovian states) using the Viterbi algorithm.

Learning concepts: Conditional probability, Markov property, stochastic optimization, dynamic programming.

Theoretical Concepts

I. **Introduction & overview**: We will consider a certain stochastic process with the following K dimensional state space $\mathbf{S} = \{s_1, s_2, ..., s_K\}$. Associated with this process is a T dimensional observation sequence $\mathbf{Y} = \{y_1, y_2, ..., y_T\}$ from amongst a possible N dimensional observation space $\mathbf{O} = \{o_1, o_2, ..., o_N\}$. Note: $y_n \in \mathbf{O}$. Further, consider an initial probability distribution given by $\mathbf{\Pi} = \{\pi_1, \pi_2, ..., \pi_K\}$. The probability transition matrix $\mathbb{P}$ is a $K \times K$ matrix with entries

$$p_{ij}(t) := \text{probability of transitioning from state } s_i \text{ to state } s_j = Prob(x_t = s_j \big| x_{t-1} = s_i),$$

and the emission matrix $\mathbb{E}$ is a $K \times N$ matrix with entries

$$e_{ij}(t) := \text{probability of observing } o_j \text{ from state } s_i = Prob(y_t = o_j \big| x_t = s_i).$$

Succinctly, we will often write $s_i \equiv i$ and $o_j \equiv j$ where it must be understood that $x_t = i$ refers to the random variable x_t taking the state s_i and $y_t = j$ refers to the random variable y_t being assigned the observable o_j. The goal of the prediction algorithm is to forecast the most likely sequence of states (events) $\mathbf{X} = \{x_1, x_2, ..., x_T\}$, where $x_n \in \mathbf{S}$ given a prescribed sequence of observables $\mathbf{Y}$, i.e. we need to compute

$$\text{argmax}_{\mathbf{X}} Prob(\mathbf{Y}, \mathbf{X}) = \text{argmax}_{\mathbf{X}} Prob(\mathbf{X} \big| \mathbf{Y}) Prob(\mathbf{Y}) = \text{argmax}_{\mathbf{X}} Prob(\mathbf{Y} \big| \mathbf{X}) Prob(\mathbf{X}).$$

Here $\text{argmax}\big(f(x)\big)$ returns the value of x at which the function $f(x)$ attains its maximum. In addition to this main goal of the project that uses the **Viterbi Algorithm**, we will also use the **Forward algorithm** to calculate the probability of a given sequence of observations. This also has useful practical implications.

Let us realize this in the form of an example. Assume that your friend is living in another country and the weather on a particular day in that country follows a *Hidden Markovian model* (**HMM**). '*Hidden*' because the Markovian states (here, the weather states) are not directly observable. For convenience, assume that there are 3 possible weather states, $\mathbf{S} = \{$rainy, cloudy, sunny$\}$, we call this the state space. Let's say, you're given what activity your friend does on a given day (which your friend will communicate to you about). This information is what you can directly observe or know. The activities your friend decides to do depends only on the type of weather on a given day. The set of all activities he does comes from the observation space which for this case we assume to be $\mathbf{O} = \{$read, shop, play$\}$. Now, based on what activity he does every day for a given number of days, you'll find the most likely sequence of the weather states (hidden states) that occurred during the given days. That is say if for 5 days his sequence of activities was $\mathbf{Y} = \{$read, read, shop, play, shop$\}$, then what's the *most likely* weather sequence for these 5 days? Viterbi algorithm would do this for you.

In addition to this task you'll also find the probability of a given sequence of your friend's activities. That is say, you want to know what is the probability that the sequence of activities of your friend for 5 consecutive days is $\mathbf{Y} = \{\text{read, read, shop, play, shop}\}$. The Forward algorithm would do this for you. While the pseudo code along with the essential calculations for the Viterbi algorithm would be given to you, you're expected to use the construction of the Forward algorithm given in this project to design a code for the same, from scratch.

II. **Construction and essential calculations of the Viterbi algorithm**: In what follows, we will fix the notation $Prob(X_1 = x_1) \equiv Prob(x_1) \equiv \pi_1$. Note that if $T = 2$, then

$$\begin{aligned}
Prob(\mathbf{Y}, \mathbf{X}) &\equiv Prob(y_1, y_2, x_1, x_2) \\
&= Prob(y_1, y_2, x_2 | x_1) Prob(x_1) \\
&= Prob(y_1, y_2 | x_2, x_1) Prob(x_2 | x_1) Prob(x_1) \\
&= Prob(y_1 | y_2, x_2, x_1) Prob(y_2 | x_2, x_1) p_{12} \pi_1 \\
&= Prob(y_1 | x_1, x_2, y_2) Prob(y_2 | x_2) p_{12} \pi_1 \\
&= Prob(y_1 | x_1) Prob(y_2 | x_2) p_{12} \pi_1 \qquad (1)
\end{aligned}$$

In general, we have

$$\begin{aligned}
Prob(\mathbf{Y}, \mathbf{X}) &\equiv Prob(\mathbf{Y} = y_1, y_2 ..., y_T, \mathbf{X} = x_1, x_2 ..., x_T) \\
&= \underbrace{Prob(x_1)}_{\pi_1} \underbrace{Prob(y_1 | x_1)}_{e_{1y_1}} \underbrace{Prob(x_2 | x_1)}_{p_{12}} \underbrace{Prob(y_2 | x_2)}_{e_{2y_2}} \cdots\cdots \underbrace{Prob(y_T | x_T)}_{e_{Ty_T}} \qquad (2)
\end{aligned}$$

The Viterbi algorithm involves *recursively* computing the Viterbi entries $V_{k,t}$

$$\begin{aligned}
V_{k,t} &:= \max Prob\big((y_1, ..., y_t), (x_1, ..., x_t = s_k)\big) \\
&= \text{probability of the best (most likely) sequence of states (ending with state } k \text{, i.e. } x_t = s_k) \\
&\quad \text{corresponding to the sequence of observables } (y_1, ..., y_t).
\end{aligned}$$

Recursive computation of $V_{k,t}$:

By comparing the terms on the right-hand side of eq. (2) and the definition of the Viterbi entries above, we see that $V_{k,t}$ can be obtained recursively and consequently using the argmax function, we can find the most likely sequence of events. The algorithm includes the calculation of the following three important terms.

- $V_{k,t} = \max\limits_{\alpha \in \mathbf{S}} \big(Prob(y_t = j | x_t = s_k) p_{\alpha k} V_{\alpha, t-1} \big) = \max\limits_{\alpha \in \mathbf{S}} \big(e_{kj} p_{\alpha k} V_{\alpha, t-1} \big)$
  with $V_{k,1} \overset{set}{=} Prob(y_1 = o_m | x_1 = s_k) \pi_k = e_{km} \pi_k$, for all $k = 1, 2, \ldots, K$ and $t = 1, 2, \ldots, T$ and
- $x_T = \underset{\alpha \in S}{\text{argmax}} \big(V_{\alpha, T} \big)$.
- x_{t-1} = back_pointer(x_t, t) = value of x_{t-1} used to compute $V_{k,t}$ for all $t = 2, 3, \ldots, T$.

Software Implementation

Pseudo-code of the Viterbi algorithm:

```
INPUT: $\mathbf{S}, \mathbf{\Pi}, \mathbb{E}, \mathbb{P}$ and $\mathbf{Y}$.
```

Part I: Initialization.

```
for each i of K states
    viterbi_prob(i,1) = $\pi_i * e_{iy_1}$
    viterbi_path(i,1) = $0$
end for
```

Part II: Compute Viterbi probabilities and Viterbi path.

```
for each j of T-1 observations starting with T=2
    for each i of K states
        viterbi_prob(i,j) = $\max_{\alpha \in \mathbf{S}} \big( e_{iy_j} * p_{\alpha i} *$viterbi_prob$(\alpha, j-1) \big)$
        viterbi_path(i,j) = $\operatorname{argmax}_{\alpha \in \mathbf{S}} \big( e_{iy_j} * p_{\alpha i} *$viterbi_prob$(\alpha, j-1) \big)$
    end for
end for
$x_T = s_{z_T}$ where $z_T := \operatorname{argmax}_{\alpha \in \mathbf{S}} \big($viterbi_prob$(\alpha, T) \big)$
```

The appearance of e_{ij} in the computation of `viterbi_path(i,j)` is unnecessary because it is non-negative and independent of α (so you may choose to skip it).

Part III: Re-tracking the most likely path $\mathbf{X}$.

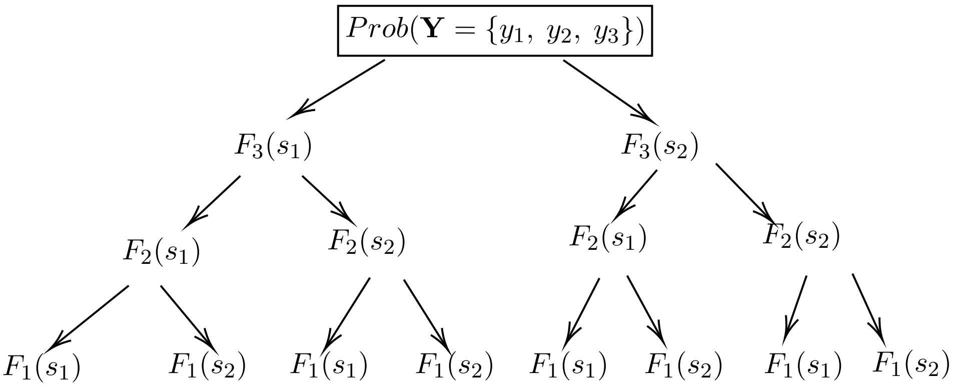
```
for each j of T-1 observations from T to 2
    $x_{j-1} = s_{z_{j-1}}$ where $z_{j-1} =$viterbi_path$(z_j, j)$
end for
```

```
OUTPUT: $\mathbf{X} = \{x_1, x_2, ..., x_T\}$, $Prob(\mathbf{Y}, \mathbf{X}) = \max_{\alpha \in \mathbf{S}} \big($viterbi_prob$(\alpha, T) \big)$
```

III. **Construction and essential calculations of the Forward algorithm**:

For convenience consider the state space $\mathbf{S} = \{s_1, s_2\}$ representing the Hidden Markov states and the observation space $\mathbf{O} = \{o_1, o_2\}$. We are interested in calculating the probability of a given observed sequence, say $\mathbf{Y} = \{y_1, y_2, y_3\}$ given the parameters of our HMM. Denote length $(\mathbf{Y}) = T$, here $T = 3$

Given the observed sequence, it's clear that there are 8 possible sequences of Markov States that can produce this sequence (since length $(\mathbf{S}) = 2$ and each observation can be associated with these 2 states). Rather than calculating the individual joint probabilities of all 8 possibilities of the state sequences and the given observed sequence and then adding them up, we use the Forward algorithm to reduce the complexity of the problem.



Here $F_t(s_k)$ denotes the probability of occurrence of the state s_k at the t^{th} position of Markov chain sequence and that sequence $\{y_1, y_2, \cdots, y_t\}$ is followed till t^{th} position.

There are three stages in the forward algorithm:

1. Initialization:

$$F_1(s_i) = \pi_i e_{iy_1}$$

For instance, in our example, we have the following initial probabilities:

$$F_1(s_1) = \pi_1 e_{1y_1} \text{ and } F_1(s_2) = \pi_2 e_{2y_1}$$

2. Recursion:

$$F_t(s_i) = \sum_{j=1}^{k} F_{t-1}(s_j) p_{ji} e_{iy_t}$$

For instance in our example, we have

$$F_2(s_1) = F_1(s_1)p_{11}e_{1y_2} + F_1(s_2)p_{21}e_{1y_2}$$

In the same way, for other values of t, $F_t(s_k)$ can be written.

3. Termination:

$$Prob(\mathbf{Y} = \{y_1, y_2, \cdots, y_T\}) = \sum_{j=1}^{k} F_T(s_j)$$

So, in our example, we are interested in

$$Prob(\mathbf{Y} = \{y_1, y_2, y_3\}) = F_3(s_1) + F_3(s_2)$$

IV. **Questions**: Implement the above algorithms on PYTHON (or MATLAB) and use your program to answer the following set of questions.

1. **Example:** In the example stated earlier on predicting the Markovian weather states assume the initial weather distribution $\mathbf{\Pi} = (0.35, 0.25, 0.4)$, the probability transition matrix $\mathbb{P} = \begin{pmatrix} 0.4 & 0.2 & 0.4 \\ 0.2 & 0.5 & 0.3 \\ 0.3 & 0.6 & 0.1 \end{pmatrix}$, where state 1 is *rainy*, state 2 is *cloudy* and state 3 is *sunny*, and the probability emission matrix $\mathbb{E} = \begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0.4 & 0.2 & 0.4 \end{pmatrix}$, where the columns (observations) are labeled in order of *reading*, *playing* and *shopping*, respectively and rows (weather states) are labeled in order of *rainy*, *cloudy* and *sunny* respectively. Now, answer the following questions:

   (a) What is the most likely weather sequence for the last 5 days given that the activities of your friend for the last 5 days are: '*read*', '*read*', '*shop*', '*play*', '*shop*'?

   (b) What is the probability that the activities of your friend for 5 consecutive days are $\mathbf{Y} = \{\text{read, read, shop, play, shop}\}$?

2. **Main Problem:** Aircraft sensor data from the Airbus A330 is used to predict the flight characteristics and accordingly modify control inputs. One such flight characteristic is pitch *up* and pitch *down* motions (observation state variables) measured by the angle of attack sensors. Any error in the pitch measurements may inadvertently affect the primary flight control laws (state variables) and have major consequences on the aerodynamic performance of the plane. In any aircraft there are three primary control laws, viz., *normal*, *alternate*, and *direct*, each of which demands distinct inputs by the pilot and the on-board flight computer system. The flight envelope and failure protection modes are also distinctly different depending on the type of control law governing the flight at any given instant, e.g., normal law may have automated low-speed anti-stall protection whereas the same may not be available while the aircraft is operated under direct law. Therefore, accurate real-time prediction of the prevailing control law is essential for continuing safe flights and is monitored carefully by the company at the Airbus engineering systems headquarters. From the Airbus database, we have the probability transition matrix $\mathbb{P}$, (where state 1 is *normal*, state 2 is *alternate* and state 3 is *direct*), probability emission matrix $\mathbb{E}$ (where the columns or observations are labeled in order of *up* and *down* and rows or aircraft control laws are labeled in order of *normal*, *alternate* and *directly* respectively) and the initial probability distribution of the aircraft control laws $\mathbf{\Pi}$.

$$\mathbb{P} = \begin{pmatrix} 0.35 & 0.45 & 0.2 \\ 0.28 & 0.32 & 0.4 \\ 0.75 & 0.1 & 0.15 \end{pmatrix}, \mathbb{E} = \begin{pmatrix} 0.1 & 0.9 \\ 0.6 & 0.4 \\ 0.7 & 0.3 \end{pmatrix} \text{ and } \mathbf{\Pi} = \{0.2,\ 0.3,\ 0.5\}$$

   Using this solve the following problems:

   (a) If the company receives the following sequence of pitch measurements at 5-minute intervals, what is the most likely sequence of aircraft control laws that would be activated during the same instant of time?
   Pitch data: '*up*', '*down*', '*down*', '*down*', '*up*', '*up*', '*up*', '*down*', '*up*', '*down*'.

   (b) What is the probability that the pitch data in eight consecutive 5-minute intervals are:

      i. $\mathbf{Y} = \{\text{up, up, up, up, up, up, up, up}\}$
      ii. $\mathbf{Y} = \{\text{down, up, down, up, down, up, down, up}\}$
      iii. $\mathbf{Y} = \{\text{up, down, up, down, up, down, up, down}\}$
      iv. $\mathbf{Y} = \{\text{up, up, down, up, down, down, down, up}\}$

v. $\mathbf{Y} = \{$down, down, down, down, down, down, down, down$\}$?

(c) From the answers obtained in $(2b)$, what possible changes can be made in the HMM parameters ($\mathbb{P}$, $\mathbb{E}$, $\mathbf{\Pi}$) such that:

i. any two observation sequences of the same length containing an equal number of *ups* and *downs* have the same probability (for example, probability of sequence $(2(b)ii)$ is equal to the probability of sequence $(2(b)iii)$? ,

ii. for a given length of an observation sequence, the probabilities of all possible observation sequences are equal?

3. (a) Consider the following emission matrix in the HMM with $\mathbb{P}$ and $\mathbf{\Pi}$ in (2):

$$\mathbb{E} = \begin{pmatrix} 0.65 & 0.35 \\ 0.82 & 0.18 \\ 0.73 & 0.27 \end{pmatrix}.$$

i. Using appropriate justification, comment on what action ('landing' or 'take-off' or 'cruising horizontally') the aircraft is most likely performing, given the HMM parameters?

ii. Now consider a new emission matrix, $\tilde{\mathbb{E}}$ by swapping the columns of E. Given that the aircraft is in the take-off action, comment on any potential anomalies in the aircraft performance based on $\tilde{\mathbb{E}}$.

(b) Given a Markovian state at a particular time instant (say, *'alternate'* at time 't', in the HMM in (2)), construct an algorithm (both theoretically and through PYTHON (or MATLAB) implementation) to compute the probability of the company receiving the following pitch data in the next eight 5-minute intervals (starting from $t_1 = t + 5, t_2 = t + 10, \cdots$).
Pitch data: '*down*', '*down*', '*down*', '*up*', '*up*', '*up*', '*down*', '*up*'
(*Note: Assume the same parameters (*$\mathbb{P}$*,* $\mathbb{E}$*,* $\mathbf{\Pi}$*) for the HMM in question* (2))

□