The idea of this document is to describe the algorithms used with HMMs for

- Prediction - Estimating the probability distribution over the state of the system at a future time, given knowledge of the state of the system now (this knowledge of the current state will be in the form of a probability distribution).

- State Estimation - Estimating the current state of the system given knowledge about the initial state, and a sequence of observations associated with time steps 1 to $t$, where $t$ is the current time step.

- Smoothing - Estimating a past state of the system (say at time $s$) given knowledge about the initial state, and a sequence of observations associated with time steps 1 to $t$, where $t$ is the current time step and $s < t$.

I will try and explain them in three ways: Using natural language, using the algorithm symbolism and using probability theory.

1 Definitions and Symbolism

The state of the system at a particular time, $t$, is a random variable, $S_t$, that can take values in the set $\{s_1, s_2, ..., s_n\}$. When used in matrix equations $S_t$ should be understood as a column vector of probability values, $[P(S_t = s_1), P(S_t = s_2), ..., P(S_t = s_n)]^T$. In words, the $i^{th}$ element of the vector $S_t$ represents the probability that the system is in state $s_i$ at time $t$. We will use double indices to represent elements of this vector, so $S_{t,i}$ will be the $i^{th}$ element of the vector $S_t$.

The system emission at a particular time, $t$, is a random variable, $E_t$, that can take values in the set $\{e_1, e_2, ..., e_m\}$.

The sequence of observations (observed emissions) is $\langle O_1, O_2, ..., O_t \rangle$, where each observation is an emission value.

Sequences of system states, emissions or observations can be indicated using colons, such that $S_{3:7}$ is the sequence of random variables $\langle S_3, S_4, S_5, S_6, S_7 \rangle$.

The transition matrix, $T$, gives the transition probabilities of the system, such that $t_{i,j} = P(S_{t+1} = s_j | S_t = s_i)$. In words, $t_{i,j}$ is the probability of the system transitioning to state $s_j$ given it is in state $s_i$. 
\[ T = \begin{bmatrix}
  t_{1,1} & t_{1,2} & \cdots & t_{1,n} \\
  t_{2,1} & t_{2,2} & \cdots & t_{2,n} \\
  \vdots & \vdots & \ddots & \vdots \\
  t_{n,1} & t_{n,2} & \cdots & t_{n,n}
\end{bmatrix} \]

The emission matrix, \( E \), gives the emission probabilities of the system, such that \( e_{i,j} = P(E_t = e_j|S_t = s_i) \). In words, \( e_{i,j} \) is the probability of observing emission \( e_j \) given the system is in state \( s_i \).

\[ E = \begin{bmatrix}
  e_{1,1} & e_{1,2} & \cdots & e_{1,m} \\
  e_{2,1} & e_{2,2} & \cdots & e_{2,m} \\
  \vdots & \vdots & \ddots & \vdots \\
  e_{n,1} & e_{n,2} & \cdots & e_{n,m}
\end{bmatrix} \]

2 Prediction - Equivalent to Markov chains

2.1 Goal

Predict the probability distribution over a future state of the system, given a probability distribution over the current state of the system.

2.2 Algorithm

Since there are no observed emissions between now and some future time, this becomes equivalent to predicting the future state of a Markov chain given a specification of its initial state.

\[ S_{t+n} = S_t^T T^n \]

2.3 Example

To see the complete details of the arithmetic calculations corresponding to this matrix equation, look at the Markov chain example at the beginning of the MarkovExamples document.

3 State Estimation - The Forward Algorithm

3.1 Goal

We want to calculate the probability distribution for the state variable at a certain time slice, \( S_t \), given the distribution over the initial state, \( S_0 \), and observations up to that point, \( O_{1:t} \).
3.2 Algorithm Idea

We use a dynamic programming algorithm to iteratively calculate the probability distribution for the state variable at times 1, 2, ..., t given the distribution of the previous state variable and the current observation.

3.3 Algorithm

Let $f_i$ be a vector of $n$ values, such that $f_t, i$ is the $f$ value associated with the $s_i$ at time $t$.

1. Let $f_0 = S_0$.

2. For $a = 1, ..., t$:
   
   (a) For $b = 1, ..., n$
       
       i. Set $f_{a,b} = \sum^n_i (f_{a-1,i}T_{i,b})E_{a,j}$, where $O_a = e_j$.

3.4 Algorithm - Natural Language Discussion

The value assigned by the forward algorithm to a state value $s_i$ at time step $t$, that is to say $f_{t,i}$, is $A \cdot B$, where:

- $A$ is the sum of the product of the $f$ value assigned to each state value at the previous time and the probability of transitioning from that previous state value to state value $s_i$.
- $B$ is the probability of the observation at time $t$, $O_t$, given $s_i$.

Since $f$ is really the unnormalized probabilities of states (see below), we are essentially looking at the transitions that will lead to each state value at time $t$, and summing over their probabilities. This makes use of the $f$-values of the previous layer and the transition probabilities. Then we multiply the result by the probability of seeing the observation at time $t$ given each state. This incorporates the observations.

3.5 Algorithm - Probability formulation

For those who are interested, we can formulate the forward algorithm directly in terms of probabilities.

- Interpretation of $f$ values: $f_t(t, i) = P(S_t = s_i, E_{1:t} = O_{1:t} | S_0) \propto P(S_t = s_i | S_0, E_{1:t} = O_{1:t})$.
- Assignment step: $P(S_t = s_i, E_{1:t} = O_{1:t} | S_0) = \sum^n_j (P(S_{t-1} = s_j, E_{1:t-1} = O_{1:t-1} | S_0)P(S_t = s_i | S_{t-1} = s_j))P(E_t = O_t | S_t = s_i)$.
3.6 Example

To see the complete details of the arithmetic calculations corresponding to this algorithm, look at the first Hidden Markov Model problem at page 29 and onwards of the MarkovExamples document.

4 Smoothing - The Forward/Backward Algorithm

4.1 Goal

We want to calculate the probability distribution for the state variable at a certain time slice, $S_s$, given the distribution over the initial state, $S_0$, and observations up to that point, $O_{1:t}$, where $s < t$. That is to say, the time slice we want the probability distribution for is in the past.

More generally, we might want to get probability distributions for every state variable, $S_s$ with $0 \leq s \leq t$, given the distribution over the initial state, $S_0$, and observations up to that point, $O_{1:t}$.

4.2 Algorithm Idea

We want to calculate $P(S_s|S_0, E_{1:t} = O_{1:t})$ for all times of interest, $s$.

Given the conditional independencies that exist between variables in a HMM, we know that:

$$P(S_s|S_0, E_{1:t} = O_{1:t}) = P(S_s|S_0, E_{1:s} = O_{1:s})P(S_s|E_{s+1:t} = O_{s+1:t})$$

So we can split the problem into two: First find the probability of each state variable at a time of interest given the initial state and observations up to and including that time. Since this is what the forward algorithm does, we already know how to do this!

Secondly, calculate the probability of a state variable at a certain time given observations after that time. This is what the backward algorithm that we introduce below will do.

We can then multiply these two results together to find the probability of a state variable at a certain time given the initial state and all observations.

4.3 Algorithm

Let us assume that we have run the forward algorithm already, and so have $f$-values for all state values at all time steps.

Let $b_t$ be a vector of $n$ values, such that $b_t,i$ is the $b$ value associated with the $s_i$ at time $t$.

1. Let $b_t$ be a vector of $n$ ones. So $b_{t,i} = 1$, for all $i$.

2. For $y = t - 1, ..., 0$:

   (a) For $z = 1, ..., n$
i. Set \( b_{y,z} = \sum_{i}^{n} (b_{y+1,i} T_{z,i} E_{y+1,j}) \), where \( O_{y+1} = e_{j} \).

Finally, for every time step, \( s \), of interest, we create an \( n \) element vector \( u_{s} \), such that \( u_{s,i} = b_{s,i} f_{s,j} \). We normalize \( u_{s} \) to get the probability distribution over the state values at time step \( s \).

### 4.4 Algorithm - Natural Language Discussion

The value assigned by the backward algorithm to a state value \( s_{i} \) at time step \( t \), that is to say \( b_{t,i} \), is the sum over \( j \) of \( A_{j} \cdot B_{j} \cdot C_{j} \), where:

- \( A_{j} \) is the \( b \) value assigned to the state value \( s_{i} \) at the next time
- \( B_{j} \) is the probability of transitioning from state value \( s_{i} \) to that next state value \( s_{j} \).
- \( C_{j} \) is the probability of the observation at time \( t+1 \), \( O_{t+1} \), given \( s_{j} \).

Since \( b \) is really the unnormalized probabilities of states (see below), we are essentially looking at the probabilities of all future observations given each possible transition out of a state value at time \( t \), and summing over these probabilities. \( B \) is the transition probabilities, \( A \) is the \( b \)-values of the transitioned to node for a given transition, and \( C \) the probability of seeing the observation at time \( t+1 \) given a potential transition.

### 4.5 Algorithm - Probability formulation

For those who are interested, we can formulate the forward algorithm directly in terms of probabilities.

- Interpretation of \( b \) values: \( b(s,i) = P(E_{s+1:t} = O_{s+1:t} | S_{s} = s_{i}) \propto P(S_{s} = s_{i} | E_{s+1:t} = O_{s+1:t}) \). The \( b \) values for the final, \( t \), time step are one, since for \( b_{t,i} \) we are trying to find the probability of all observations after time \( t \) given \( S_{t} = s_{i} \), of which there are none. Formally we would need to define the symbols \( E_{t+1:t} \) and \( O_{t+1:t} \) as both being the empty set.

- Assignment step: \( P(E_{s+1:t} = O_{s+1:t} | S_{s} = s_{i}) = \sum_{j}^{n} (P(S_{s+1} = s_{j} | S_{s} = s_{i})P(E_{s+1} = O_{s+1} | S_{s+1} = s_{j})P(E_{s+1:t} = O_{s+1:t} | S_{s} = s_{j})) \)

### 4.6 Example

To see the complete details of the arithmetic calculations corresponding to this algorithm, look at the second Hidden Markov Model problem at page 63 and onwards of the *MarkovExamples* document.