CS 520: Temporal Estimation

16:198:520
Instructor: Wes Cowan

1 Formulation

Recall the basic hidden Markov model that we are dealing with: we have an underlying state sequence $X_0, X_1, X_2, \ldots$ describing the evolution of some model over time. We take the initial distribution of states $P(X_0)$ to be known, and additionally we take the transition model to be known, i.e., $P(X_{t+1}|X_0, X_1, X_2, \ldots, X_t)$, the distribution of the state at time $t+1$ given the prior history, is known. We take a *Markov* assumption, in that we assume that the state at time $t+1$ depends *only* on the state at time t, that is

$$P(X_{t+1}|X_0, X_1, \ldots, X_t) = P(X_{t+1}|X_t). \tag{1}$$

It is worth saying again explicitly: while we will ultimately take the sequence of states as unobserved, the underlying *model*, in terms of the initial distribution and transition probabilities, is known.

We equip this transition model with an observation model. That is, at any time t, we collect a data point Y_t that is some function of the underlying state. In particular, we will assume that

$$P(Y_t|X_0, X_1, \ldots, X_t, Y_1, Y_2, \ldots, Y_{t-1}) = P(Y_t|X_t), \tag{2}$$

which is to say that the observation at time t, while it *could* depend on all previous results and observations, depends only on the current state.

These allow for a convenient factoring of the joint probability of some sequence of events as

$$P(X_0, X_1, Y_1, X_2, Y_2, \ldots, X_t, Y_t) = P(X_0)\prod_{t=1}^{n} P(X_t|X_{t-1})P(Y_t|X_t). \tag{3}$$

We will focus primarily on the following four problems: Filtering, Prediction, Smoothing, and Most Likely Estimation.

2 Filtering

The question filtering asks is: given a sequence of observations up to some time t, what can we say about the distribution of state X_t? In other words, given everything that we've seen so far, what should we believe about the actual state of the model? It is convenient to define the following function:

$$\text{Belief}_t(x) = P(X_t = x|Y_1, Y_2, \ldots, Y_t). \tag{4}$$

The function $\text{Belief}_t(x)$ represents the probability that the system is in state x at time t, given all the observed data thus far. We will derive an efficient means for computing this function recursively. In particular, consider trying to

express the belief state at time $t+1$ in terms of the belief state at time t. This leads to the following expansion,

$$\begin{aligned}
\text{Belief}_{t+1}(x) &= P(X_{t+1} = x | Y_1, Y_2, \ldots, Y_t, Y_{t+1}) \\
&= \beta P(X_{t+1} = x, Y_{t+1} | Y_1, Y_2, \ldots, Y_t) \textit{ some conditional manipulation} \\
&= \beta \sum_{x'} P(X_t = x', X_{t+1} = x, Y_{t+1} | Y_1, Y_2, \ldots, Y_t) \textit{ marginalization} \\
&= \beta \sum_{x'} P(X_t = x' | Y_1, \ldots, Y_t) P(X_{t+1} = x | Y_1, \ldots, Y_t, X_t = x') P(Y_{t+1} | Y_1, \ldots, Y_t, X_t = x', X_{t+1} = x) \\
&= \beta \sum_{x'} P(X_t = x' | Y_1, \ldots, Y_t) P(X_{t+1} = x | X_t = x') P(Y_{t+1} | X_{t+1} = x) \\
&= \beta \sum_{x'} \text{Belief}_t(x') P(X_{t+1} = x | X_t = x') P(Y_{t+1} | X_{t+1} = x)
\end{aligned} \tag{5}$$

It is convenient perhaps to factor that in the following way:

$$\text{Belief}_{t+1}(x) = \beta \left(\sum_{x'} \text{Belief}_t(x') P(X_{t+1} = x | X_t = x') \right) P(Y_{t+1} | X_{t+1} = x), \tag{6}$$

which makes it clear that in order to end up at state x at time $t+1$, first we had to arrive at state x *from some state* x', and then once we were at state x we observed data Y_{t+1}.

If $\text{Belief}_t(x)$, as a function over all possible states x, represents the total knowledge available at time t, the above (subject to the normalization factor β) represents this prior belief accurately pushed forward one step based on new data. We consider all the places we might have been previously, and how likely were were to be in each of them, and consider what state they could give rise to in the next step, constrained by the data we actually collected.

This is a very efficient algorithm that can be done in a dynamic, online fashion, as more data is collected. And notice the additional efficiency as well provided by the fact that you do not need to track your prior beliefs - at any time, Belief_t captures the full state of your probabilistic knowledge of the system, and you have no need to keep track of Belief_{t-1} as well.

One important thing to consider in this case: how would we modify this analysis if our data points were only intermittent, i.e., at some points we had effectively $Y_t = \textbf{null}$ which told us nothing?

One final note: in any recursive formulation, we must address the base case, which allows us to kick-start our recursive formulation. So what is the base case here? Prior to collecting any data, at time $t = 0$ all we have to rely on is our knowledge of the initial distribution, which gives us

$$\text{Belief}_0(x) = P(X_0 = x). \tag{7}$$

3 Prediction

The question addressed by prediction is this: given data collected up to time t, what can we say about the state of the system k steps in the future? What can we anticipate about the likely trajectories of the system? It is convenient to introduce the following notation:

$$\text{Predict}_t(x, k) = P(X_{t+k} = x | Y_1, Y_2, \ldots, Y_t). \tag{8}$$

To derive an efficient means of computing this, we can again consider a recursive formulation, utilizing conditioning and marginalization to manipulate the underlying probabilities:

$$\begin{aligned}
\text{Predict}_t(x, k+1) &= P(X_{t+k+1} = x | Y_1, Y_2, \ldots, Y_t) \\
&= \sum_{x'} P(X_{t+k} = x', X_{t+k+1} = x | Y_1, Y_2, \ldots, Y_t) \\
&= \sum_{x'} P(X_{t+k} = x' | Y_1, Y_2, \ldots, Y_t) P(X_{t+k+1} = x | Y_1, Y_2, \ldots, Y_t, X_{t+k} = x') \\
&= \sum_{x'} P(X_{t+k} = x' | Y_1, Y_2, \ldots, Y_t) P(X_{t+k+1} = x | X_{t+k} = x') \\
&= \sum_{x'} \text{Predict}_t(x', k) P(X_{t+k+1} = x | X_{t+k} = x')
\end{aligned} \tag{9}$$

In this case, we are essentially computing the predicted distribution at time $t+k+1$ by predicting the state at time $t+k$, and pushing our knowledge forward one step via our knowledge of the transition model. This is very similar to filtering, but we do not have intermediate data to improve the state of our knowledge. We are only factoring what we know about the transition model into our knowledge of the future.

What is the base case for starting our recursion? In the case case, we are considering $k = 0$, which is to say we are forecasting *no* steps in the future. Indeed, we get

$$\text{Predict}_t(x, 0) = P(X_t = x | Y_1, Y_2, \ldots, Y_t) = \text{Belief}_t(x), \tag{10}$$

that is, prediction starts by taking our current total belief state (as the total expression of all our current knowledge).

4 Smoothing

The question smoothing addresses is that at any time k, given the data we collected *up to* time k and the data we collected *after* time k, what can we say about the state of the system *at* time k? This is distinct from the filtering case as we are combining information from both directions - not only does the past tell us where we are going, but the present tells us where we must have been. It is convenient to introduce the following function for $1 \leq k < t$,

$$\text{Smooth}_t(x, k) = P(X_k = x | Y_1, \ldots, Y_t). \tag{11}$$

It is convenient to analyze this probability by splitting it into the forward part (data prior to k informing the state at time k) and the backward part (data post k informing what the state at time k *must have been*).

$$\begin{aligned}
&\text{Smooth}_t(x, k) \\
&= \alpha P(X_k = x, Y_{k+1}, \ldots, Y_t | Y_1, Y_2, \ldots, Y_k) \\
&= \alpha \sum_{x'} P(X_k = x, X_{k+1} = x', Y_{k+1}, \ldots, Y_t | Y_1, Y_2, \ldots, Y_k) \\
&= \alpha \sum_{x'} P(X_k = x | Y_1, \ldots, Y_k) P(X_{k+1} = x' | X_k = x, Y_1, \ldots, Y_k) P(Y_{k+1}, \ldots, Y_t | Y_1, Y_2, \ldots, Y_k, X_k = x, X_{k+1} = x') \\
&= \alpha \sum_{x'} P(X_k = x | Y_1, \ldots, Y_k) P(X_{k+1} = x' | X_k = x) P(Y_{k+1}, \ldots, Y_t | X_{k+1} = x')
\end{aligned} \tag{12}$$

The above splits the problem of smoothing into a 'forward' part, effectively filtering information from time 1 to k to inform the state at time k, and a 'backwards' part, looking at how likely a given state was to give rise to all the

data collected after that point. These two parts are joined by the transition model. It is convenient to define the following function:

$$\text{Backwards}_t(x,k) = P(Y_k,\ldots,Y_t|X_k = x), \tag{13}$$

so that

$$\text{Smooth}_t(x,k) = \alpha \sum_{x'} \text{Belief}_k(x) P(X_{k+1} = x'|X_k = x)\text{Backwards}_t(x', k+1). \tag{14}$$

It remains to compute the Backwards function. Again, we marginalize and condition, as needed:

$$\begin{aligned}
\text{Backwards}_t(x,k) &= P(Y_k,\ldots,Y_t|X_k = x) \\
&= \sum_{x'} P(X_{k+1} = x', Y_k,\ldots,Y_t|X_k = x) \\
&= \sum_{x'} P(X_{k+1} = x'|X_k = x)P(Y_k|X_k = x, X_{k+1} = x')P(Y_{k+1},\ldots,Y_t|X_k = x, X_{k+1} = x', Y_k) \\
&= \sum_{x'} P(X_{k+1} = x'|X_k = x)P(Y_k|X_k = x)P(Y_{k+1},\ldots,Y_t|X_{k+1} = x') \\
&= \sum_{x'} P(X_{k+1} = x'|X_k = x)P(Y_k|X_k = x)\text{Backwards}_t(x', k+1).
\end{aligned} \tag{15}$$

Note that this recursion actually goes in the opposite direction we've seen so far (hence the name *Backwards*). In this case, the base case is actually

$$\text{Backwards}_t(x,t) = P(Y_t|X_t = x), \tag{16}$$

which is read directly from the observation model.

Smoothing is definitely the most computationally intensive of the processes we've looked at so far, as it must be processed in both directions at once. This makes it hard to do an online version of smoothing - at every step, the new data may re-inform your understanding at every prior step.

5 Most Likely Estimation - Viterbi's Algorithm

One key point about the topics of the previous sections was that they were all *pointwise* estimates, looking at the state of the model at a given time. We may additionally be interested in the entire trajectory of the model: given a set of data $Y_1,\ldots,Y_t$, what was the most likely sequence of states $X_1,\ldots,X_t$? A classic example of this is the problem of doing text-to-speech, in which case you want a most likely reconstruction of a whole sentence, rather than individual distributions for certain words or sound segments.

In this case, we are looking for the *most likely estimate* or

$$\text{MLE}(t) = \text{argmax}_{x_1,x_2,\ldots,x_t} P(X_1 = x_1,\ldots,X_t = x_t|Y_1,\ldots,Y_t), \tag{17}$$

where argmax denotes that we are looking for the sequence that achieves the maximum value of the probability.

To compute this, we introduce the following intermediary problem:

$$\text{MLE}_t(x) = \max_{x_1,x_2,\ldots,x_{t-1}} P(X_1 = x_1, X_2 = x_2,\ldots,X_t = x|Y_1,\ldots,Y_t). \tag{18}$$

The value $\text{MLE}_t(x)$ is the maximum likelihood of any sequence that terminates with $X_t = x$. This we can compute

recursively, in the following way:

$$\begin{aligned}
&\text{MLE}_{t+1}(x) \\
&= \max_{x_1,x_2,\ldots,x_t} P(X_1 = x_1, X_2 = x_2, \ldots, X_t = x_t, X_{t+1} = x | Y_1, \ldots, Y_t, Y_{t+1}) \\
&= \max_{x_1,x_2,\ldots,x_t} \alpha\ P(X_1 = x_1, X_2 = x_2, \ldots, X_t = x_t, X_{t+1} = x, Y_{t+1} | Y_1, \ldots, Y_t) \\
&= \max_{x_1,x_2,\ldots,x_t} \alpha\ P(X_1 = x_1, X_2 = x_2, \ldots, X_t = x_t | Y_1, \ldots, Y_t) P(X_{t+1} = x | X_t = x_t) P(Y_{t+1} | X_{t+1} = x) \\
&= \max_{x_t} \alpha\ \text{MLE}_t(x_t) P(X_{t+1} = x | X_t = x_t) P(Y_{t+1} | X_{t+1} = x)
\end{aligned} \tag{19}$$

Disregarding (briefly) the normalization factor α, this formula has a natural interpretation - the maximum likelihood associated with terminating at x at time $t+1$ can be thought of in terms of the most likely ways of arriving at time t, transitioning to x under the transition model, and collecting the corresponding observation Y_t. Of all these possibilities, taking the maximum over all possible x_t yields the most likely path.

This suggests a natural computational approach: compute $\text{MLE}_0(x) = P(X_0 = x)$ for all states x, then iteratively compute $\text{MLE}_k(x)$ for all states x in terms of the previous MLE_{k-1}. The final result will be a table of likelihood values associated with ending at any state x at time t. The most likely termination state is therefore $x_t^* = \text{argmax}_x \text{MLE}_t(x)$. But what of the rest of the most likely sequence?

Computationally, this can be approached by tracking at every step, for every state, *the previous state that maximized the likelihood to the current state*; that is, every time the maximum is computed for MLE_k, track the state that gave the maximum. Once this table of pointers has been established, the most likely sequence can be reconstructed by starting at the most likely terminal state, and referencing the most likely previous state. The memory requirements for this process can be non-trivial, but ultimately unavoidable (in full generality) because new data can always dramatically shift the assessment of previously assessed state sequences.