EE378A Statistical Signal Processing Lecture 10 - 04/28/2016

Lecture 10: State Estimation of Hidden Markov Process

Lecturer: Jiantao Jiao, Tsachy Weissman Scribe: Dominic Delgado, Lewis Guignard, Greg Kehoe

In this lecture, we complete and expand the recursive algorithm for state estimation of a hidden markov process through a memoryless noisy channel.

1 Recap

Let there be a Markov Process $\{X_n\}_{n\geq 1}$ sent through a memoryless channel, and measurements $\{Y_n\}_{n\geq 1}$. Then the stochastic system is characterized by:

$$P_{x_1}, \{P_{X_t|X_{t-1}}\}_{t\geq 2}, \{P_{Y_t|X_t}\}_{t\geq 1}$$

Let

$$\alpha_t(X_t) = p(x_t|y^t)$$
$$\beta_t(X_t) = p(x_t|y^{t-1})$$
$$\gamma_t(X_t) = p(x_t)|y^n$$

In lecture 9, we found that:

$$\alpha_t(X_t) = \frac{\beta_t(X_t)p(y_t|x_t)}{\sum_{\tilde{x}_t}\beta_t(X_t)p(y_t|\tilde{x}_t)}$$
$$\beta_{t+1}(X_t) = \sum_{x_t}\alpha_t(x_t)p(x_{t+1}|x_t)$$
$$\gamma_t(X_t) = \sum_{X_{t+1}}\frac{\gamma_{t+1}\alpha_{t+1}p(x_{t+!}|x_t)}{\beta_{t+1}(X_{t+1})}$$

2 Finding $P(x^n|y^n)$

Until now, we have been considering the posterior distribution of a single X_t, but in certain settings, e.g. Communications, it may be useful to consider the posterior distribution of the whole sequence X^n.

First, let us prove that the state of a HMP conditioned on the observations Y^n is still Markov.

Claim:

$$X^{t-1} - (X_t, Y^n) - X^n_{t+1}$$

Proof:

$$\begin{aligned}
P(x^n, y^n) &= \prod_{i=1}^{n} p(x_i|x_{i-1})p(y_i|x_i) \\
&= \prod_{i=1}^{t} p(x_i|x_{i-1})p(y_i|x_i) \prod_{i=t+1}^{n} p(x_i|x_{i-1})p(y_i|x_i)
\end{aligned}$$

These two terms can be seen as two general functions:

$$\begin{aligned}
&= \phi_1(x^t, y^t)\phi_2(x_t^n, y_{t+1}^n) \\
&= \phi_1(x^{t+1}, x_t, y^n)\phi_2(x_t, x_{t+1}^n, y^n)
\end{aligned}$$

Here we have trivially expanded Y^t and Y_{t+1}^n to go from 1 to n, since this includes any subset of Y^n. Recall that the Markov chain $X - Y - Z$ holds iff $p(x, y, z) = \phi_1(x, y)\phi_2(y, z)$. Therefore $X^{t-1} - (X_t, Y^n) - X_{t+1}^n$.

Now we are ready to derive the posterior probability of the sequence X^n. Using the above lemma, we may expand the probability as follows:

$$\begin{aligned}
p(x^n|y^n) &= p(x_n|y^n)p(x_{n-1}|x_n, y^n)p(x_{n-2}|x_{n-1}, y^n), \dots, p(x_1|x_2, y^n) \\
&= p(x_n|y^n) \prod_{t=1}^{n-1} p(x_t|x_{t+1}, y^t)
\end{aligned}$$

We let $y^n \to y^t$ because we already have clean state information x_{t+1}

$$\begin{aligned}
p(x_t|x_{t+1}, y^t) &= \frac{p(x_t, x_{t+1}|y^t)}{p(x_{t+1}|y^t)} = \frac{p(x_t|y^t)p(x_{t+1}|x_t, y^t)}{p(x_{t+1}|y^t)} \\
&= \frac{\alpha_t(x_t)p(x_{t+1}|x_t)}{\beta_{t+1}(x_{t+1})} \\
\Rightarrow p(x^n|y^n) &= \gamma_n(x_n) \prod_{t=1}^{n-1} \frac{\alpha_t(x_t)p(x_{t+1}|x_t)}{\beta_{t+1}(x_{t+1})}
\end{aligned}$$

Maximum Likelihood Estimation of X^n

$$\begin{aligned}
p(x^n|y^n) &= \gamma_n(x_n) \prod_{t=1}^{n-1} \frac{\alpha_t(x_t)p(x_{t+1}|x_t)}{\beta_{t+1}(x_{t+1})} \\
\log p(x^n|y^n) &= \sum_{t=1}^{n} g_t(x_t, x_{t+1})
\end{aligned}$$

Where we define g as:

$$g_t(x_t, x_{t+1}) := log\frac{\alpha_t(x_t)p(x_{t+1}|x_t)}{\beta_{t+1}(x_{t+1})}, \quad t \in \{1, \dots, n-1\}$$
$$g_t(x_t, x_{t+1}) := log\gamma_n(x_n), \quad t = n$$

If g were a function of x_t only, we could simply use a greedy algorithm and maximize each term. Since g_t is a function of both x_t, x_{t+1}, we can be *almost* greedy.

What we want:

$$X_{ML} = \arg\max_{x^n} \sum_{t=1}^{n} g_t(x_t, x_{t+1})$$

Definition 1. *for* $1 \leq k \leq n$*, Let*

$$M_k(x_k) := \max_{x_{k+1}^n} \sum_{t=k}^{n} g_t(x_t, x + t + 1)$$

Then,

$$\begin{aligned} M_k(x_k) &= \max_{x_{k+1}} \max_{x_{k+2}^n} (g_k(x_k, x_{k+1}) + \sum_{t=k+1}^{n} g_t(x_t, x_{t+1})) \\ &= \max_{x_{k+1}} (g_k(x_k, x_{k+1}) + \max_{x_{k+2}^n} \sum_{t=k+1}^{n} g_t(x_t, x_{t+1})) \\ &= \max_{x_{k+1}} (g_k(x_k, x_{k+1}) + M_{k+1}(x_{k+1})) \end{aligned}$$

Since $g(x_n, x_{n+1}) = g(x_n) = log\gamma_n(x_n)$ only depends on one term, x_n, we may first compute $M_n(x_n)$, and that allows us to start computing terms that depend on two variables (M_{n-1}, etc.). This recursive computation is sometimes called the Viterbi Algorithm.

3 Bellman Equations, aka Viterbi Algorithm

The Bellman Equations, referred to in the communication setting as the Viterbi Algorithm, is an application of a technique called dynamic programming. Using the recurrence equations derived in the previous section, we can express the Viterbi Algorithm as follows.

```
1: function VITERBI
2:     $M_n(x_n) \leftarrow log\gamma_n(x_n)$                                ▷ Initialization
3:     for $1 \leq k \leq n$ do
4:         $M[k] \leftarrow M_k(x_k) = \max_{x_{k+1}}(g_k(x_k, x_{k+1}) + M_{k+1}(x_{k+1}))$
5:     end for
6:     $M \leftarrow \max_{x_1} M_1(x_1)$                        ▷ Maximum Likelihood of Sequence
7: end function
```

Note that the maximizing sequence $(X_1^{ML}, X_2^{ML}, ...X_n^{ML}) = argmax_{x^n} P(X^n|Y^n))$, where

$$X_1^{ML} = argmax_{X_1} M_1(X_1), X_2^{ML} = argmax_{X_2} M_2(X_2), ...$$
$$X_{k+1}^{ML} = argmax_{X_{k+1}}(g_k(x_k, x_{k+1}) + M_{k+1}(X_{k+1}))$$

Maximizing Sequence:

$$(x_1^{ML}, x_2^{ML}, \dots, x_n^{ML})$$
$$\text{where: } x_1^{ML} = \arg\max_{x_1} M(x_1)$$
$$\text{and: } x_{k+1}^{ML} = \arg\max_{x_{k+1}}(g_k(x_k, x_{k+1}) + M_{k+1}(x_{k+1})$$

4 Reflections

4.1 Optimality Conditions

Let us distinguish between the maximizing the likelihood of the whole sequence versus maximizing the likelihood of getting each individual bit correct. Consider the random vector X^n with the following distribution:

$$X^n = \begin{cases} 1111...11 & w.p. 0.1 \\ 1000...00 & w.p. \frac{1}{n} 0.9 \\ 0100...00 & w.p. \frac{1}{n} 0.9 \\ 0010...00 & w.p. \frac{1}{n} 0.9 \\ ... \end{cases}$$

Clearly the ML sequence is the all-one sequence, but $P(X_i = 1) = 0.1 + \frac{0.9}{n} < P(X_i) = 0$, so if the goal is to guess the whole sequence correctly, we should guess all ones, but if the goal is to make as few bit errors as possible on average, we should guess all zeros.

4.2 Reflection on forward - backards recursion

Consider:

$$X \sim p_x \rightarrow \boxed{p_{Y|X}} \rightarrow Y$$
$$p_{X|Y}(x|y) = \frac{p_X(x) p_{Y|X}(y|x)}{\sum_{\tilde{x}} p_{Y|X}(y|x)}$$
$$= F(p_X, p_{Y|X}, y)$$

$F(p_X, p_{Y|X}, y)$ outputs a vector.

We can also write F as

$$F(p_X, p_{Y|X}) = \sum_{\tilde{x}} p_x(\tilde{x}) p_{Y|X}(y|\tilde{x}) = G(p_X, p_{Y|X})$$

which corresponds to the matrix for the entire distribition $P_{x|y}$.

4.3 Recursions redefined

Forward Recursions redefined

$$\alpha_t = F(\beta_t, p_{Y_t|X_t}, y_t) \tag{1}$$

$$\beta_{t+1} = G(\alpha_t, p_{x_{t+1}|x_t})) \tag{2}$$

Backward Recursions redefined

$$\gamma_t = G(\gamma_{t+1}, F(\alpha_t, P_{x_{t+1}|x_t})) \tag{3}$$