The Baum-Welch & Viterbi Algorithms

Filtering equations for the class of fully-discrete HMMs are relatively simple to derive through Bayesian manipulation of the posteriors. These discrete algorithms are interesting because they embody the most powerful elements of HMM theory in a very simple framework. The methods are readily-implementable and have become the workhorse in applied areas where machine learning algorithms are needed. The algorithms for filtering, smoothing and parameter estimation are analogous to their counterparts in continuous models, but the theoretical background required for understanding is minimal in the discrete setting.

1 Equations for Filtering, Smoothing & Prediction

Let n denote a discrete time, and suppose that X_n is an unobserved Markov chain taking values in a discrete state-space denoted by $\mathcal{S}$. Let Λ denote X_n's kernel of transition probabilities so that

$$\mathbb{P}(X_{n+1} = x) = \sum_{v \in \mathcal{S}} \Lambda(x|v)\mathbb{P}(X_n = v)$$

for any $x \in \mathcal{S}$, and $\mathbb{P}(X_0 = x) = p_0(x)$.

Noisy measurements are taken in the form of a process Y_n which is a nonlinear function of X_n, plus some noise,

$$Y_n = h(X_n) + W_n$$

where W_n is an iid Gaussian random variable with mean zero and variance $\gamma^2 > 0$. The main feature of this discrete model is the **memoryless-channel** which allows the process to 'forget the past':

$$\mathbb{P}(Y_n, X_n = x|X_{n-1} = v, Y_{0:n-1}) = \mathbb{P}(Y_n|X_n = x)\Lambda(x|v)$$

for any $n \geq 0$ and for all $x, v \in \mathcal{S}$.

1.1 Filtering

The filtering mass function is

$$\pi_n(x) \doteq \mathbb{P}(X_n = x|Y_{0:n})$$

for all $x \in \mathcal{S}$. Through an application of Bayes rule along with the properties of the HMM, we are able to break down π_n as follows,

$$\pi_n(x) = \frac{\mathbb{P}(X_n = x, Y_{0:n})}{\mathbb{P}(Y_{0:n})}$$

$$= \frac{\sum_{v \in \mathcal{S}} \mathbb{P}(Y_n, X_n = x | X_{n-1} = v, Y_{0:n-1}) \mathbb{P}(X_{n-1} = v, Y_{0:n-1})}{\mathbb{P}(Y_{0:n})}$$

$$= \frac{\mathbb{P}(Y_n | X_n = x) \sum_{v \in \mathcal{S}} \mathbb{P}(X_n = x | X_{n-1} = v) \mathbb{P}(X_{n-1} = v, Y_{0:n-1})}{\mathbb{P}(Y_{0:n})}$$

$$= \frac{\mathbb{P}(Y_n | X_n = x) \sum_{v \in \mathcal{S}} \Lambda(x|v) \mathbb{P}(X_{n-1} = v | Y_{0:n-1})}{\mathbb{P}(Y_n | Y_{0:n-1})}$$

$$= \frac{\mathbb{P}(Y_n | X_n = x) \sum_{v \in \mathcal{S}} \Lambda(x|v) \pi_{n-1}(v)}{\sum_{x \in \mathcal{S}} \text{numerator}}$$

where the memoryless-channel allows for the conditioning that occurs between the second and third lines. This recursive breakdown of the filtering mass is the **forward Baum-Welch Equation**, and can be written explicitly for the the system with Gaussian observation noise

$$\pi_n(x) = \frac{1}{c_n} \psi_n(x) \sum_{v \in \mathcal{S}} \Lambda(x|v) \pi_{n-1}(v) \tag{1}$$

where c_n is a normalizing constant, and ψ_n is a likelihood function

$$\psi_n(x) \doteq \mathbb{P}(Y_n | X_n = x) = \exp\left\{ -\frac{1}{2} \left(\frac{Y_n - h(x)}{\gamma} \right)^2 \right\}.$$

Equation (1) is convenient because it keeps the distribution updated without having to recompute old statistics as new data arrives. In 'real-time' it is efficient to use this algorithm to keep track of X's latest movements, but older filtering estimates will not be optimal after new data has arrived. The smoothing distribution must be used to find the optimal estimate of X at some time in the past.

1.2 Smoothing

For some time $N > n$ up to which data has been collected, the smoothing mass function is

$$\pi_{n|N}(x) \doteq \mathbb{P}(X_n = x | Y_{0:N}).$$

Through an application of Bayes rule along with the properties of the model, the smoothing mass can be written as follows,

$$\pi_{n|N}(x) = \frac{\mathbb{P}(Y_{n+1:N}|X_n = x)\pi_n(x)}{\mathbb{P}(Y_{n+1:N}|Y_{0:n})}$$

$$= \frac{\sum_{v\in\mathcal{S}} \mathbb{P}(Y_{n+1:N}|X_{n+1} = v, X_n = x)\Lambda(v|x)\pi_n(x)}{\mathbb{P}(Y_{n+1:N}|Y_{0:n})}$$

$$= \frac{\sum_{v\in\mathcal{S}} \mathbb{P}(Y_{n+2:N}|X_{n+1} = v)\psi_{n+1}(v)\Lambda(v|x)\pi_n(x)}{\mathbb{P}(Y_{n+2:N}|Y_{0:n+1})\mathbb{P}(Y_{n+1}|Y_{0:n})}$$

$$= \frac{\sum_{v\in\mathcal{S}} \mathbb{P}(Y_{n+2:N}|X_{n+1} = v)\psi_{n+1}(v)\Lambda(v|x)\pi_n(x)}{\mathbb{P}(Y_{n+2:N}|Y_{0:n+1})c_{n+1}}. \qquad (*)$$

where c_{n+1} is the normalizing constant from equation (1). Now suppose that we define a likelihood function for the events after time n,

$$\alpha_n^N(x) = \frac{\mathbb{P}(Y_{n+1:N}|X_n = x)}{\mathbb{P}(Y_{n+1:N}|Y_{0:n})}$$

for $n < N$ with the convention that $\alpha_N^N \equiv 1$. Then the smoothing mass can be written as the product of the filtering mass with α

$$\pi_{n|N}(x) = \alpha_n^N(x)\pi_n(x)$$

and from $(*)$ we can see that α_n^N is given recursively by a **backward Baum-Welch Equation**

$$\alpha_n^N(x) = \frac{1}{c_{n+1}} \sum_{v\in\mathcal{S}} \alpha_{n+1}^N(v)\psi_{n+1}(v)\Lambda(v|x). \qquad (2)$$

Clearly, computation of the smoothing distribution requires a computation of all filtering distribution up to time N followed by the backward recursion to compute α^N. In exchange for doing this extra work, the sequence of X's estimates will suggest a path taken by X that is more plausible than the path suggested by the filtering estimates.

1.3 Prediction

The prediction distribution is easier to compute than smoothing. For $n < N$, the prediction distribution is

$$\pi_{N|n}(x) \doteq \mathbb{P}(X_N = x|Y_{0:n})$$

and is merely computed by extrapolating the filtering distribution,

$$\pi_{N|n}(x) = \sum_{v\in\mathcal{S}} \Lambda(x|v)\pi_{N-1|n}(v) = \sum_{v\in\mathcal{S}} \Lambda^{N-n}(x|v)\pi_n(v)$$

where Λ^{N-n} denotes the transition probability over $N-n$ time steps.

If X_n is a positive recurrent Markov chain, then there is an invariant and the prediction distribution will converge to as $N \to \infty$. In some cases, the rate at which this convergence occurs will be proportional to the spectral gap in Λ.

Suppose X_n can take one of m-many finite-state, and is a recurrent Markov chain with only 1 communication class. Let $\Lambda \in \mathbb{R}^{m\times m}$ be the matrix of transition probabilities for X, and suppose that $\Lambda_{ji} > 0$ so that

$$\mathbb{P}(X_{n+1} = x_i|X_n = x_j) = \Lambda_{ji} > 0$$

for all $i, j \leq m$. Then the prediction distribution is

$$\pi_{N|n} = \pi_n\Lambda^{N-n}$$

and will converge exponentially fast to the invariant measure with a rate proportional to the second eigenvalue of Λ. To see why this is true, consider the basis of eigenvectors $(\mu_i)_{i\leq m}$ of Λ, some of which may be generalized,

$$\mu_{i+1}(\Lambda - \beta_i I) = \mu_i$$

for some $i \geq 1$. Assuming that μ_1 is the unique invariant mass function of X_n, we have $\mu_1\Lambda = \mu_1$. By the Perron-Frobenius Theorem we can sort the eigenvalues so that $1 = \beta_1 > |\beta_2| \geq |\beta_3| \geq \ldots \geq |\beta_m|$, and we know that β_1 is a simple root of the characteristic polynomial and therefore μ_1 is not a generalized eigenvector. From here we can see that

$$-\frac{1}{k}\log\|\pi_n\Lambda^k - \mu_1\| = -\frac{1}{k}\log\|(\pi_n - \mu_1)\Lambda^k\| = -\frac{1}{k}\log\|(a_1\mu_1 + a_2\mu_2 + \ldots a_m\mu_m)\Lambda^k\|$$

$$= -\frac{1}{k}\log\|a_1\mu_1 + a_2\beta_2^k\mu_2 + \ldots a_m\mu_m\Lambda^k\| \sim \frac{1}{k}\log\left(1 + a_2'|\beta_2^k|\right) \sim |\beta_2|$$

as $k \to \infty$. The spectral gap of Λ is $1 - |\beta_2|$, and from the convergence rate we see that a greater spectral gap means that the prediction distribution will take less time to converge to the invariant measure. In general, the Perron-Frobenius theorem can be applied to a recurrent finite-state Markov chain provided that there is some integer $k < \infty$ for which $\Lambda_{ji}^k > 0$ for all $i, j \leq m$.

2 Baum-Welch Algorithm for Learning Parameters

It is not very realistic to assume that we have apriori knowledge of the HMM that is completely accurate. However, stationarity of X means that we are observed repeated behavior of X, albeit through noisy measurements, but nevertheless we should be able to judge the frequencies with which X occupies parts of the state-space and the frequencies with which it moves about.

If we have already computed the smoothing distribution based on a model that is 'close' in some sense, then we should have

$$\frac{1}{N}\sum_{n=1}^{N}\mathbb{P}(X_n = x|Y_{0:N}) \approx \mu(x) \tag{3}$$

$$\frac{1}{N}\sum_{n=1}^{N}\mathbb{P}(X_n = x, X_{n-1} = v|Y_{0:N}) \approx \Lambda(x|v)\mu(v) \tag{4}$$

where μ is the stationary law of X. With the Baum-Welch algorithm, we can in fact employ some optimization techniques to find a sequence of model estimates which are of increasing likelihood, and it turns out that the (3) and (4) are similar to the optimal improvement in selecting the sequence of models.

Consider two model parameters θ and θ'. The Baum-Welch algorithm uses the Kullback-Leibler divergence to compare the two models,

$$0 \le D(\theta\|\theta') = \sum_{\vec{x}\in\mathcal{S}^{N+1}} \frac{\mathbb{P}^{\theta}(X_{0:N} = \vec{x}, Y_{0:N})}{\mathbb{P}^{\theta}(Y_{0:N})} \log\left(\frac{\mathbb{P}^{\theta}(X_{0:N} = \vec{x}, Y_{0:N})\mathbb{P}^{\theta'}(Y_{0:N})}{\mathbb{P}^{\theta'}(X_{0:N} = \vec{x}, Y_{0:N})\mathbb{P}^{\theta}(Y_{0:N})}\right)$$

$$= \log\left(\frac{\mathbb{P}^{\theta'}(Y_{0:N})}{\mathbb{P}^{\theta}(Y_{0:N})}\right) + \sum_{\vec{x}\in\mathcal{S}^{N+1}} \frac{\mathbb{P}^{\theta}(X_{0:N} = \vec{x}, Y_{0:N})}{\mathbb{P}^{\theta}(Y_{0:N})} \log\left(\frac{\mathbb{P}^{\theta}(X_{0:N} = \vec{x}, Y_{0:N})}{\mathbb{P}^{\theta'}(X_{0:N} = \vec{x}, Y_{0:N})}\right).$$

If we set

$$Q(\theta\|\theta') \doteq \sum_{\vec{x}\in\mathcal{S}^{N+1}} \mathbb{P}^{\theta}(X_{0:N} = \vec{x}, Y_{0:N}) \log\left(\mathbb{P}^{\theta'}(X_{0:N} = \vec{x}, Y_{0:N})\right),$$

we then have a simplified expression,

$$0 \le D(\theta\|\theta') = \log\left(\frac{\mathbb{P}^{\theta'}(Y_{0:N})}{\mathbb{P}^{\theta}(Y_{0:N})}\right) + \frac{Q(\theta\|\theta) - Q(\theta\|\theta')}{\mathbb{P}^{\theta}(Y_{0:N})}$$

and rearranging the inequality we have

$$\frac{Q(\theta\|\theta') - Q(\theta\|\theta)}{\mathbb{P}^{\theta}(Y_{0:N})} \le \log\left(\frac{\mathbb{P}^{\theta'}(Y_{0:N})}{\mathbb{P}^{\theta}(Y_{0:N})}\right),$$

from which we see that $Q(\theta\|\theta') > Q(\theta\|\theta)$ implies that θ' has greater likelihood than θ. The Baum-Welch algorithm uses this inequality as the basis for a criteria to iteratively refine the estimated model parameter. The algorithm obtains a sequence $\{\theta^\ell\}_\ell$ for which $Q(\theta^{\ell-1}\|\theta^\ell) \geq 0$, and so their likelihoods are increasing but bounded,

$$\mathbb{P}^{\theta^{\ell-1}}(Y_{0:N}) \leq \mathbb{P}^{\theta^\ell}(Y_{0:N}) \leq \mathbb{P}^{\hat{\theta}^{mle}}(Y_{0:N}),$$

where $\hat{\theta}^{mle}$ is the maximum likelihood estimate of θ. Therefore, $\{\theta^\ell\}_\ell$ will have a limit at θ^* such that

$$\mathbb{P}^{\theta^*}(Y_{0:N}) = \lim_\ell \mathbb{P}^{\theta^\ell}(Y_{0:N}),$$

but it may be the case that $\mathbb{P}^{\theta^*}(Y_{0:N}) < \mathbb{P}^{\hat{\theta}^{mle}}(Y_{0:N})$ (see figure 1).

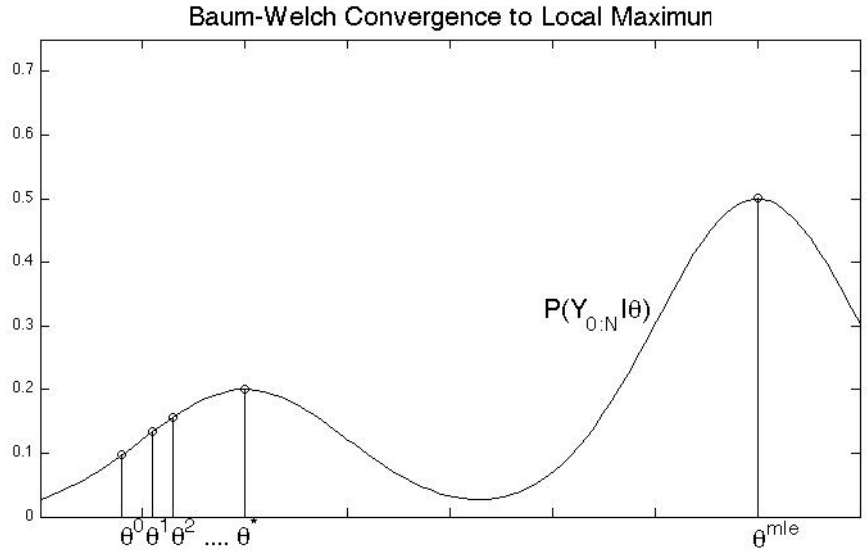

Figure 1: A sequence of Baum-Welch parameter estimates with increasing likelihood, but the sequence is caught at a local maximum.

In doing computations, a maximum (perhaps only a local maximum) of $Q(\theta\|\ \cdot\ )$ needs to be found. First-order conditions are good technique for finding one, and using the HMM we can expand $Q(\theta\|\theta')$ into an explicit form,

$$Q(\theta\|\theta') = \sum_{\vec{x}\in\mathcal{S}^{N+1}} \mathbb{P}^\theta(X_{0:N} = \vec{x}, Y_{0:N}) \left\{ \psi_0^{\theta'}(\vec{x}_0) p_0^{\theta'}(\vec{x}_0) + \sum_{n=1}^N \log\left(\psi_n^{\theta'}(\vec{x}_n)\Lambda^{\theta'}(\vec{x}_n|\vec{x}_{n-1})\right)\right\}, \tag{5}$$

from which we see that it is possible to differentiate with respect to θ', add the Lagrangians, and then solve for the optimal model estimate.

The Baum-Welch algorithm is equivalent to the expectation-maximization (EM) algorithm; the EM algorithm maximizes the expectation of the log-likelihood function which is equivalent to maximizing Q,

$$\theta^\ell = \arg\max_\theta \mathbb{E}^{\theta^{\ell-1}} \left[\log \left(\mathbb{P}^\theta(Y_{0:N}, X_{0:N}) \right) \middle| Y_{0:N} \right] = \arg\max_\theta Q(\theta^{\ell-1} \| \theta).$$

2.1 Model Re-Estimation for Parametric Transition Probabilities

Suppose that $X_n \in \mathbb{Z}$, with transition probabilities parameterized by $\theta \in (0, \infty)$ so that

$$\mathbb{P}(X_{n+1} = i | X_n = j) = \frac{1}{c(\theta)} \exp\{-\theta |i-j|^2\}, \qquad \forall i, j \in \mathbb{Z},$$

where $c(\theta) = \sum_{i=-\infty}^{\infty} \exp\{-\theta|i-j|^2\}$. Ignoring the parts that do not depend on θ', the log-likelihood is

$$Q(\theta \| \theta') = -\sum_{n=1}^{N} \mathbb{E}^\theta \left[\theta' |X_n - X_{n-1}|^2 + \log c(\theta') \middle| \mathcal{F}_N^Y \right],$$

and if we differentiate with respect to θ' we have the following first-order conditions,

$$\frac{\partial}{\partial \theta'} Q(\theta \| \theta') = -\sum_{n=1}^{N} \mathbb{E}^\theta \left[|X_n - X_{n-1}|^2 - \frac{\sum_i |i-j|^2 \exp\{-\theta|i-j|^2\}}{c(\theta')} \middle| \mathcal{F}_N^Y \right] = 0$$

for any $j \in \mathbb{Z}$. The solution to the first-order conditions is θ' that satisfies

$$\mathbb{E}^{\theta'} \left[|X_1 - X_0|^2 \middle| X_0 = j \right] = \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}^\theta \left[|X_n - X_{n-1}|^2 \middle| \mathcal{F}_N^Y \right]$$

for any j.

2.2 Model Re-Estimation for Finite-State Markov Chains

Suppose $X_n \in \mathcal{S} = \{1, \ldots, m\}$, so that

$$\mathbb{P}(X_{n+1} = i | X_n = j) = \Lambda_{ji}$$

for all $i, j \in \mathcal{S}$. We will look for a sequence $\Lambda^{(\ell)}$ which maximizes $Q(\Lambda^{(\ell-1)} \| \ \cdot \ )$ subject to the constraints $\sum_i \Lambda_{ji} = 1$ for all $j \le m$. Letting δ_j be the Lagrange multiplier for the jth constraint, the first order conditions are then,

$$\frac{\partial}{\partial \Lambda_{ji}} \left(Q(\Lambda^{(\ell-1)} \| \Lambda) - \delta_j \sum_r \Lambda_{jr} \right) = \frac{\partial}{\partial \Lambda_{ji}} Q(\Lambda^{(\ell-1)} \| \Lambda) - \delta_j = 0. \qquad (**)$$

Multiplying by Λ_{ji} and summing over i the expression in $(**)$ becomes

$$0 = \sum_i \Lambda_{ji} \left(\frac{\partial}{\partial \Lambda_{ji}} Q(\Lambda^{(\ell-1)} \| \Lambda) - \delta_j \right) = \sum_i \Lambda_{ji} \frac{\partial}{\partial \Lambda_{ji}} Q(\Lambda^{(\ell-1)} \| \Lambda) - \delta_j$$

which means $\delta_j = \sum_i \Lambda_{ji} \frac{\partial}{\partial \Lambda_{ji}} Q(\Lambda^{(\ell-1)} \| \Lambda)$. By multiplying $(**)$ by Λ_{ji} and then rearranging terms it is found that the optimal $\Lambda_{ji}^{(\ell)}$ must be chosen among the set of Λ's such that

$$\Lambda_{ji} = \frac{\Lambda_{ji} \frac{\partial}{\partial \Lambda_{ji}} Q(\Lambda^{(\ell-1)} \| \Lambda)}{\sum_r \Lambda_{jr} \frac{\partial}{\partial \Lambda_{jr}} Q(\Lambda^{(\ell-1)} \| \Lambda)}. \tag{6}$$

Now, using the expansion in (5), the derivative of $Q(\Lambda^{(\ell-1)} \| \Lambda)$ with respect to Λ_{ji} can be computed as follows:

$$\frac{\partial}{\partial \Lambda_{ji}} Q(\Lambda^{(\ell-1)} \| \Lambda) = \mathbb{E} \left[\frac{\partial}{\partial \Lambda_{ji}} \log \mathbb{P}(Y_{0:N}, X_{0:N} | \Lambda) \Big| Y_{0:N}, \Lambda^{(\ell-1)} \right]$$

$$= \mathbb{E} \left[\sum_{n=1}^{N} \frac{1}{\Lambda_{ji}} \mathbf{1}_{\{X_n = i, X_{n-1} = j\}} \Big| Y_{0:N}, \Lambda^{(\ell-1)} \right] = \frac{1}{\Lambda_{ji}} \sum_{k=1}^{N} \mathbb{P}(X_n = i, X_{n-1} = j | Y_{0:N}, \Lambda^{(\ell-1)})$$

and by plugging this into equation (6) it is easily seen that the solution is

$$\Lambda_{ji}^{(\ell)} = \frac{\sum_{n=1}^{N} \mathbb{P}(X_n = i, X_{n-1} = j | Y_{0:N}, \Lambda^{(\ell-1)})}{\sum_i \text{numerator}} \tag{7}$$

where $\mathbb{P}(X_n = i, x_{n-1} = j | Y_{0:N}, \Lambda^{(\ell-1)}) = \alpha_n^N(i) \psi_n(i) \Lambda_{ji}^{(\ell-1)} \pi_{n-1}(j)$. It also happens that equation (7) enforces non-negativity of Λ_{ji}, which is required for well-posedness of the algorithm. Equation (7) is equivalent to the estimates that were conjectured in (3) and (4).

3 The Viterbi Algorithm

Sometimes it may be more important to estimate the entire path of X. The Viterbi algorithm applies the properties of the HMM along with dynamic programming to find an optimal sequence $\widehat{V}_{0:N} \in \mathcal{S}^{N+1}$ that maximizes the joint-posterior probability

$$\widehat{V}_{0:N} = (\widehat{V}_0, \ldots, \widehat{V}_N) \doteq \arg \max_{\vec{x} \in \mathcal{S}^{N+1}} \mathbb{P}(X_{0:N} = \vec{x}, Y_{0:N}).$$

Given the data $Y_{0:N}$, smoothing can be used to 'look-back' and make estimates of X_n for some $n < N$, but neither equations (1) or (2) is a joint posterior, meaning that they will not be able to tells us the posterior probability of a path $\vec{x} \in \mathcal{S}^{N+1}$. The size of our problem

would grow exponentially with N if we needed to compute the posterior distribution of $X's$ paths, but the Viterbi algorithm allows us to obtain the MAP estimator of X's path with without actually calculating the posterior probabilities of all paths.

The memoryless channel of the HMM allows us to write the maximization over paths as a nested maximization,

$$\max_{\vec{x}\in\mathcal{S}^{N+1}} \mathbb{P}(X_{0:N}=\vec{x}, Y_{0:N}) = \max_{v\in\mathcal{S}} \psi_N(v) \max_{\vec{x}\in\mathcal{S}^N} \Lambda(v|\vec{x}_{N-1})\mathbb{P}(X_{0:N-1}=\vec{x}, Y_{0:N-1})$$

$$= \psi_N(\widehat{V}_N) \max_{\vec{x}\in\mathcal{S}^N} \Lambda(\widehat{V}_N|\vec{x}_{N-1})\mathbb{P}(X_{0:N-1}=\vec{x}, Y_{0:N-1}), \qquad (\dagger)$$

where ψ is the likelihood and c_N is the normalizing constant, both from the forward Baum-Welch equation in (1). To take advantage of this nested structure, it helps to define the following recursive function,

$$\begin{aligned} \phi_0(v) &\doteq \psi_0(v)\mathbb{P}(X_0=v) \\ \phi_n(v) &\doteq \psi_n(v) \max_x \Lambda(v|x)\phi_{n-1}(x), \qquad \text{for } n=1,2,3,\ldots,N. \end{aligned}$$

We then place ϕ is the nested structure of $(\dagger)$ and work backwards to obtain the optimal path,

$$\begin{aligned} \widehat{V}_N &= \arg\max_v \phi_N(v) \\ \widehat{V}_n &= \arg\max_v \Lambda(\widehat{V}_{n+1}|v)\phi_n(v), \qquad \text{for } n=N-1,N-2,\ldots,2,1,0 \end{aligned}$$

thus obtaining the optimal path in $O(N)$-many computations. It would have taken $O\left(|\mathcal{S}|^N\right)$-many computations to obtain the posterior distribution of the paths.

We are interested in the Viterbi algorithm mainly because the path of estimates returned by the filtering and smoothing may

Remark 1. *The unnormalized probabilities in ϕ quickly fall below machine precision levels, so it is better to consider a logarithmic version of Viterbi,*

$$\begin{aligned} \log\phi_0(v) &= \log\psi_0(v) + \log\mathbb{P}(X_0=v) \\ \log\phi_n(v) &= \log\psi_n(v) + \max_x \{\log\Lambda(v|x) + \log\phi_{n-1}(x)\} \end{aligned}$$

and the use the $\log\phi_n$'s in the dynamic programming step,

$$\begin{aligned} \widehat{V}_N &= \arg\max_v \log\phi_N(v) \\ \widehat{V}_n &= \arg\max_v \left\{\log\Lambda(\widehat{V}_{n+1}|v) + \log\phi_n(v)\right\}, \qquad \textit{for } n=N-1,N-2,\ldots,2,1,0. \end{aligned}$$

References

[1] Y. Bar-Shalom, X. R. LI, Estimation and Tracking: Principles, Techniques and Software. Boston: Artech House, 1993.

[2] L. Baum, T. Petrie, G. Soules, N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," The Annals of Mathematical Statistics, Vol. 41, No. 1 (Feb., 1970), pp. 164-171.

[3] L. Baum, L. Welch, "Statistical Estimation Procedure for Probabilistic Functions of Finite Markov processes." Submitted for publication Proc. Nat. Acad. Sci. USA.

[4] A.H. Jazwinski, Stochastic Processes and Filtering Theory. Academic Press, New York, 1970.

[5] Y.A. Kutoyants. Statistical Inference for Ergodic Diffusion Processes. Springer, London, 2004

[6] L. R. Rabiner "A tutorial on Hidden Markov Models and selected applications in speech recognition". Proceedings of the IEEE 77 (2), (February 1989). pages: 257-286.

[7] B. Rozovsky, A. Petrov, "Optimal Nonlinear Filtering for Track-Before-Detect in IR Image Sequences." SPIE proceedings: Signal and Data Processing of Small Targets, vol. 3809, Denver, Co, 1999.

[8] B. L. Rozovskii, A. Petrov, R. B. Blazek, "Interacting Banks of Bayesian Matched Filters." SPIE Proceedings: Signal and Data Processing of Small Targets, Vol. 4048, Orlando, FL, 2000.