Supplementary Note: Detailed Computations

March 3rd 2017

Taehee Lee

1. Forward Algorithm:

$$\mathbf{F}^{\boldsymbol{i}}\left(\boldsymbol{n},\boldsymbol{t}_{\boldsymbol{n-1}}^{\boldsymbol{i}},\boldsymbol{t}_{\boldsymbol{n}}^{\boldsymbol{i}}\right) \triangleq p\left(v_{1:n}^{i},w_{1:n}^{i},\mathrm{A}_{n-1}^{i}=t_{n-1}^{i},\mathrm{A}_{n}^{i}=t_{n}^{i}\middle|d_{1:\mathrm{L}_i}^{i}\right)$$

$$=\sum_{t} p\left(v_{1:n}^{i},w_{1:n}^{i},\mathrm{A}_{n-2}^{i}=t,\mathrm{A}_{n-1}^{i}=t_{n-1}^{i},\mathrm{A}_{n}^{i}=t_{n}^{i}\middle|d_{1:\mathrm{L}_i}^{i}\right)$$

$$=\sum_{t} p(v_{n}^{i},w_{n}^{i}|\mathrm{A}_{n}^{i}=t_{n}^{i})p\left(v_{1:n-1}^{i},w_{1:n-1}^{i},\mathrm{A}_{n-2}^{i}=t,\mathrm{A}_{n-1}^{i}=t_{n-1}^{i},\mathrm{A}_{n}^{i}=t_{n}^{i}\middle|d_{1:\mathrm{L}_i}^{i}\right)$$

$$=\sum_{t} p(v_{n}^{i}|\mathrm{A}_{n}^{i}=t_{n}^{i})p(w_{n}^{i}|\mathrm{A}_{n}^{i}=t_{n}^{i})p\left(\mathrm{A}_{n}^{i}=t_{n}^{i}\middle|\mathrm{A}_{n-2}^{i}=t,\mathrm{A}_{n-1}^{i}=t_{n-1}^{i},d_{n-2}^{i},d_{n-1}^{i},d_{n}^{i}\right)p\left(v_{1:n-1}^{i},w_{1:n-1}^{i},\mathrm{A}_{n-2}^{i}=t,\mathrm{A}_{n-1}^{i}=t_{n-1}^{i}\middle|d_{1:\mathrm{L}_i}^{i}\right)$$

$$=\sum_{t} \mathrm{N}\left(v_{n}^{i}\middle|\tau^{i}+\mu_{t_{n}^{i}},\sigma_{t_{n}^{i}}\right)f\left(w_{n}^{i}\middle|\theta_{t_{n}^{i}}\right)\rho\left(\frac{t_{n-1}^{i}-t}{d_{n-1}^{i}-d_{n-2}^{i}},\frac{t_{n}^{i}-t_{n-1}^{i}}{d_{n}^{i}-d_{n-1}^{i}}\right)\mathrm{F}^{i}\left(n,t,t_{n-1}^{i}\right)$$

$$=\mathbf{N}\left(\boldsymbol{v}_{\boldsymbol{n}}^{\boldsymbol{i}}\middle|\boldsymbol{\tau}^{\boldsymbol{i}}+\boldsymbol{\mu}_{\boldsymbol{t}_{\boldsymbol{n}}^{\boldsymbol{i}}},\boldsymbol{\sigma}_{\boldsymbol{t}_{\boldsymbol{n}}^{\boldsymbol{i}}}\right)\boldsymbol{f}\left(\boldsymbol{w}_{\boldsymbol{n}}^{\boldsymbol{i}}\middle|\boldsymbol{\theta}_{\boldsymbol{t}_{\boldsymbol{n}}^{\boldsymbol{i}}}\right)\sum_{\boldsymbol{t}}\boldsymbol{\rho}\left(\frac{\boldsymbol{t}_{\boldsymbol{n-1}}^{\boldsymbol{i}}-\boldsymbol{t}}{\boldsymbol{d}_{\boldsymbol{n-1}}^{\boldsymbol{i}}-\boldsymbol{d}_{\boldsymbol{n-2}}^{\boldsymbol{i}}},\frac{\boldsymbol{t}_{\boldsymbol{n}}^{\boldsymbol{i}}-\boldsymbol{t}_{\boldsymbol{n-1}}^{\boldsymbol{i}}}{\boldsymbol{d}_{\boldsymbol{n}}^{\boldsymbol{i}}-\boldsymbol{d}_{\boldsymbol{n-1}}^{\boldsymbol{i}}}\right)\mathbf{F}^{\boldsymbol{i}}\left(\boldsymbol{n},\boldsymbol{t},\boldsymbol{t}_{\boldsymbol{n-1}}^{\boldsymbol{i}}\right)$$

2. Backward Sampling:

$$\mathbb{P}\left(\mathbf{A}_{\boldsymbol{n}}^{\boldsymbol{i,l}}=\boldsymbol{t}\middle|\mathbf{A}_{\boldsymbol{n+1}}^{\boldsymbol{i,l}}=\tilde{\boldsymbol{t}}_{\boldsymbol{n+1}}^{\boldsymbol{i,l}},\mathbf{A}_{\boldsymbol{n+2}}^{\boldsymbol{i,l}}=\tilde{\boldsymbol{t}}_{\boldsymbol{n+2}}^{\boldsymbol{i,l}},\boldsymbol{v}_{\boldsymbol{1:n+2}}^{\boldsymbol{i}},\boldsymbol{w}_{\boldsymbol{1:n+2}}^{\boldsymbol{i}},\boldsymbol{d}_{\boldsymbol{1:\mathrm{L}_i}}^{\boldsymbol{i}}\right)$$

$$\propto \mathbb{P}\left(\mathrm{A}_{n}^{i,l}=t,\mathrm{A}_{n+1}^{i,l}=\tilde{t}_{n+1}^{i,l},\mathrm{A}_{n+2}^{i,l}=\tilde{t}_{n+2}^{i,l},v_{1:n+2}^{i},w_{1:n+2}^{i}\middle|d_{1:\mathrm{L}_i}^{i}\right)$$

$$=p\left(v_{n+2}^{i},w_{n+2}^{i}\middle|\mathrm{A}_{n+2}^{i,l}=\tilde{t}_{n+2}^{i,l}\right)p\left(\mathrm{A}_{n}^{i,l}=t,\mathrm{A}_{n+1}^{i,l}=\tilde{t}_{n+1}^{i,l},\mathrm{A}_{n+2}^{i,l}=\tilde{t}_{n+2}^{i,l},v_{1:n+1}^{i},w_{1:n+1}^{i}\middle|d_{1:\mathrm{L}_i}^{i}\right)$$

$$=p\left(v_{n+2}^{i}\middle|\mathrm{A}_{n+2}^{i,l}=\tilde{t}_{n+2}^{i,l}\right)p\left(w_{n+2}^{i}\middle|\mathrm{A}_{n+2}^{i,l}=\tilde{t}_{n+2}^{i,l}\right)p\left(\mathrm{A}_{n+2}^{i,l}=\tilde{t}_{n+2}^{i,l}\middle|\mathrm{A}_{n}^{i,l}=t,\mathrm{A}_{n+1}^{i,l}=\tilde{t}_{n+1}^{i,l},d_{n}^{i},d_{n+1}^{i},d_{n+2}^{i}\right)p\left(\mathrm{A}_{n}^{i,l}=t,\mathrm{A}_{n+1}^{i,l}=\tilde{t}_{n+1}^{i,l},v_{1:n+1}^{i},w_{1:n+1}^{i}\middle|d_{1:\mathrm{L}_i}^{i}\right)$$

$$=\mathrm{N}\left(v_{n+2}^{i}\middle|\tau^{i}+\mu_{\tilde{t}_{n+2}^{i,l}},\sigma_{\tilde{t}_{n+2}^{i,l}}\right)f\left(w_{n+2}^{i}\middle|\theta_{\tilde{t}_{n+2}^{i,l}}\right)\rho\left(\frac{\tilde{t}_{n+1}^{i,l}-t}{d_{n+1}^{i}-d_{n}^{i}},\frac{\tilde{t}_{n+2}^{i,l}-\tilde{t}_{n+1}^{i,l}}{d_{n+2}^{i}-d_{n+1}^{i}}\right)\mathrm{F}^{i}\left(n+1,t,\tilde{t}_{n+1}^{i,l}\right)\propto\boldsymbol{\rho}\left(\frac{\tilde{\boldsymbol{t}}_{\boldsymbol{n+1}}^{\boldsymbol{i,l}}-\boldsymbol{t}}{\boldsymbol{d}_{\boldsymbol{n+1}}^{\boldsymbol{i}}-\boldsymbol{d}_{\boldsymbol{n}}^{\boldsymbol{i}}},\frac{\tilde{\boldsymbol{t}}_{\boldsymbol{n+2}}^{\boldsymbol{i,l}}-\tilde{\boldsymbol{t}}_{\boldsymbol{n+1}}^{\boldsymbol{i,l}}}{\boldsymbol{d}_{\boldsymbol{n+2}}^{\boldsymbol{i}}-\boldsymbol{d}_{\boldsymbol{n+1}}^{\boldsymbol{i}}}\right)\mathbf{F}^{\boldsymbol{i}}\left(\boldsymbol{n}+\mathbf{1},\boldsymbol{t},\tilde{\boldsymbol{t}}_{\boldsymbol{n+1}}^{\boldsymbol{i,l}}\right)$$

3. What we will need to do next with the existing Matlab codes:

We do not have the distribution of radiocarbon data, which is f in the above formulations, now. Therefore, we first ask it of Alan Jones, who is a graduate student of Professor Lisiecki. Anyway, at least we can expect that, for each dataset i and n, $f(w_n^i|\theta_t) > 0$ for only small portion of t by truncating in a compact set, which helps us to compute the forward sums faster than the previous setting only using $\delta^{18}\mathrm{O}$ data.

Note that, if we also truncate the domain of the transition function ρ and set zero for the rest of them: for example, we can consider $\rho(r_1, r_2) > 0$ if and only if both r_1 and r_2 lies in the closed interval $[0.25,4]$, and set 0 outside of it. I.e., in computing the forward algorithm, we do not need to consider all possible combinations of (t_{n-1}^i, t_n^i), but it is enough to consider the cases satisfying the following:

$$f\left(w_n^i \middle| \theta_{t_n^i}\right) > 0, \qquad \frac{t_{n-1}^i - t}{d_{n-1}^i - d_{n-2}^i} \in [0.25,4], \qquad \frac{t_n^i - t_{n-1}^i}{d_n^i - d_{n-1}^i} \in [0.25,4]$$

1. We only need to consider a small portion of $t_n^i = m$ which satisfies the first condition for each n, and just assign zeros to the forward terms for other $t_n^i = m$'s.

2. For a fixed t_n^i, the third condition implies that we only need to consider a small portion of $t_{n-1}^i = s$ lying in $\left[t_n^i - 4\left(d_n^i - d_{n-1}^i\right), t_n^i - 0.25\left(d_n^i - d_{n-1}^i\right)\right]$.

3. For fixed t_n^i and t_{n-1}^i, the second condition suggests us to consider only a small portion of t lying in $\left[t_{n-1}^i - 4\left(d_{n-1}^i - d_{n-2}^i\right), t_{n-1}^i - 0.25\left(d_{n-1}^i - d_{n-2}^i\right)\right]$.

To exploit the first observation efficiently, I think that it is not enough to replace the precomputation table of emission $\log \mathrm{N}\left(v_n^i \middle| \tau^i + \mu_{t_n^i}, \sigma_{t_n^i}\right)$ with $\log \mathrm{N}\left(v_n^i \middle| \tau^i + \mu_{t_n^i}, \sigma_{t_n^i}\right) + \log f\left(w_n^i \middle| \theta_{t_n^i}\right)$ only. Instead, what about inserting a 'if' statement, such that if the input value of t_n^i does not lie in the compact support of f then simply return zero (or $-\infty$ if the algorithm works in the log scale), in the Matlab function for computing forward algorithm?

In computing the backward sampling, we do also not need to continue to sample for all possible previously sampled values of $(\tilde{t}_{n+1}^{i,l}, \tilde{t}_{n+2}^{i,l})$, but it is enough to consider the cases satisfying the following:

$$f\left(w_{n+2}^i \middle| \theta_{\tilde{t}_{n+2}^{i,l}}\right) > 0, \qquad \frac{\tilde{t}_{n+1}^{i,l} - t}{d_{n+1}^i - d_n^i} \in [0.25,4], \qquad \frac{\tilde{t}_{n+2}^{i,l} - \tilde{t}_{n+1}^{i,l}}{d_{n+2}^i - d_{n+1}^i} \in [0.25,4]$$

1. If the input $\tilde{t}_{n+2}^{i,l}$ does not satisfy the first condition, break the current sampling and go back to the very first step to start a new sampling.

2. If the inputs $\tilde{t}_{n+1}^{i,l}$ and $\tilde{t}_{n+2}^{i,l}$ do not satisfy the third condition, break the current sampling and go back to the very first step to start a new sampling.

3. Suppose that the first and third condition were satisfied. Then we do not need to compute the probability mass function for all $t \leq \tilde{t}_{n+1}^{i,l}$: instead, just compute the terms for t's satisfying the second condition: $t \in \left[\tilde{t}_{n+1}^{i,l} - 4\left(d_{n+1}^i - d_n^i\right), \tilde{t}_{n+1}^{i,l} - 0.25\left(d_{n+1}^i - d_n^i\right)\right]$.