Brief Primer on HMM-Match

Luan Lin[1], Deborah Khider[2], Lorraine E. Lisiecki[2], Charles E. Lawrence[1]

[1]Division of Applied Mathematics, Brown University, Providence RI 02912, USA

[2]Department of Earth Science, University of California, Santa Barbara, Santa Barbara, CA 93106, USA

This primer briefly describes the HMM-Match algorithm for alignment of an input record to a stack of proxy values.

Overview

Hidden Markov models (HMM) including HMM-Match have three components,

1) Generative model (Primer Section I): HMM models are called generative models because they simulate the events that led to the data. Specifically HMM-Match simulates underlying geophysical and geochemical sedimentation events that generated the proxy data.

   $$\text{Sedimentation Events} \rightarrow \text{Data}$$

   Because all the model's assumptions must be made mathematically explicit they must be well defined. The most important goal of this primer is to highlight HMM-Match's assumptions and explain how they are represented mathematically.

2) An algorithm to infer the alignment (Primer Section II): The goal of HMM-Match is to infer common times of sediment deposition in two records, i.e. to infer alignment of the two records. To draw inferences on common times of deposition from the data using a generative model requires reversal of the logic

   $$\text{Data} \rightarrow \text{Sedimentation events (Alignment)}$$

HMM-Match, like other generative statistical inference procedures, employs Bayes rule to achieve this reversal. Because all the alignment variables of HMM-Match are interrelated, the application of Bayes rule is somewhat complex and involves two stages. In this primer we seek to communicate the central concepts of this component of HMM-Match. Because the results returned in this component are mathematical consequences of the generative model, all the assumptions employed to draw the desired inferences are made in the formulation of the generative model.

3) Parameter estimation (Primer Section III): All but two of the parameters of HMM-Match were obtained from independent sources, as described in the paper. We employ the well-established but complex Baum-Welch expectation maximization (EM) algorithm to estimate the two remaining parameters using the data in the input record. This section gives a brief overview of the Baum-Welch algorithm. A classic tutorial and a text book give a comprehensive description the Baum-Welch algorithm(1, 2). A mathematical description the specific EM employed by HMM-Match is presented in the supplement.

I. Generative Model:

1) Naturally there are variations in sedimentation rates over time and space. To account for these variations, records based on stratigraphic core samples must be aligned. The Match algorithm optimally aligns paleoclimate proxies between individual records using a deterministic dynamic programming algorithm. Of course there is uncertainty in such alignments; HMM-Match is designed to quantify this uncertainty. HMM-Match simulates the physical sedimentation processes that lead to the observed data. Since the purpose of this primer is to explain the central concepts of the HMM-Match model some aspects peripheral aspects of the algorithm, including those concerning the positioning of the ends of input records, are left to the supplement.

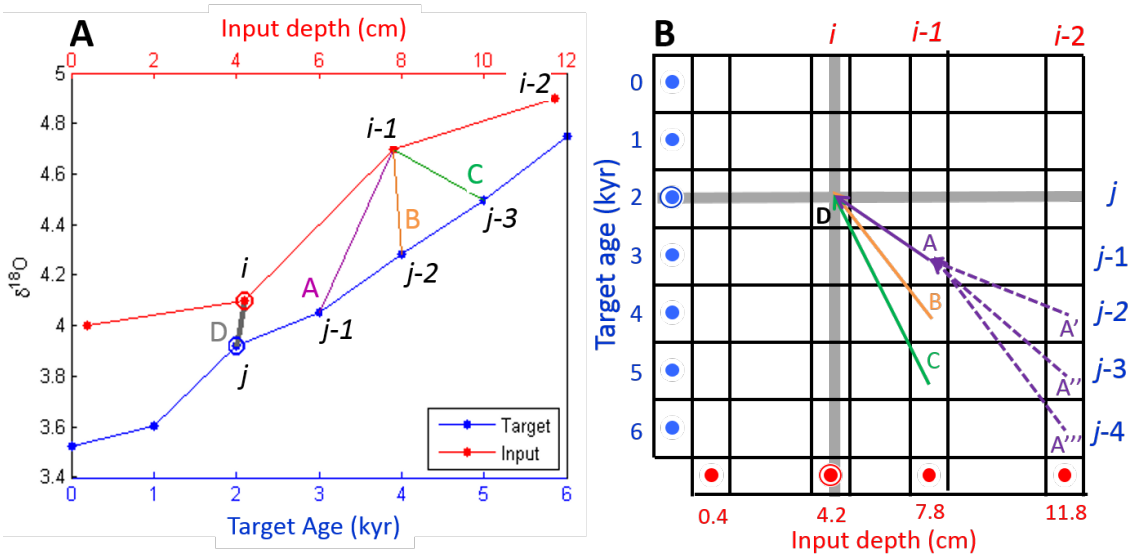

Figure 1. Several possible ways to align one portion of an input $\delta^{18}O$ record to a target $\delta^{18}O$ record. (a) The input $\delta^{18}O$ record (red) plotted versus depth (top axis) and the target $\delta^{18}O$ record (blue) plotted versus age (bottom axis). Lines labeled A, B, C, and D indicate potential alignment tie points between the two records as described in this primer. $\delta^{18}O$ measurements in the input record are labeled relative to point i, and points in the target are labeled relative to point j. (b) The same potential alignments (A-D) plotted as target age versus input depth. Here, the slopes of the colored lines correspond to different alignment ratios between points i and i-1. Dotted purple lines (labeled A′, A′′, and A′′′) indicate different possible alignment ratios between points i-1 and i-2.

2) To assess alignment uncertainty, HMM-Match uses a probabilistic model. We begin by defining a vector of variables $\mathbf{A} = (A_1, A_2, ..., A_n)$, where A_i is the unknown point in the stack that was deposited at the same time as the ith point in the input, where to follow the sedimentation process the index i=1 corresponds to the point at the physical bottom of the core and i = n to the point at the top of the core.

Mathematically, $A_i = j$ if the ith point in the input was deposited at the same time as point j in the stack, i.e. the ith point in the input should be aligned to the jth point in the stack. Because we are uncertain about this alignment $\mathbf{A}$ is a random variable. Accordingly, the model assigns probabilities to the random variables $\mathbf{A} = (A_1, A_2, ..., A_n)$. To make this more concrete we use the example shown in Figure 1

to illustrate the probabilistic alignment of the ith input data point, visualized both as input and stack $\delta^{18}O$ values versus their respective depth/time (Figure 1a) and as input depth versus stack time (Figure 1b). Specifically, HMM-Match must account for all alignments of the four data points in the input record to the 6 1-kyr intervals in the target. This description works up-core so that it follows the sedimentation process. Figure 1 illustrates one possible alignment of the input $\delta^{18}O$ value at 4.2 cm (index i) to stack $\delta^{18}O$ value with an age of 2 kyr (index j). For this alignment $A_i = 2$, see label D in Figure 1. All other points in this alignment matrix are modeled in the same way.

3) HMM-Match, like its Match predecessor, assumes that each point in the input (and stack) is shallower and younger than the point below it. So in our example if $A_1 = j$ and $A_2 = j'$ then $j < j'$. While deposition of ith input point depends on the history of the all sediments deposited before it, if we knew the time of deposition of the point just below it, i.e. point i-1, then we only need to account for the additional sediment that was deposited between points i and point i-1. Figure 1 shows three possible alignments of the previous input point, i-1, at a depth of 7.8 cm:
$A_{i-1} = 3, A_{i-1} = 4,$ or $A_{i-1} = 5$, labeled as A, B, and C respectively.

HMM-Match captures this incremental sedimentation process by modeling the age of the ith point based on the age of the (i-1)st point using a Markov chain. In a Markov chain all variables are dependent on all the other variables, but this dependence on all the other variables is entirely through its nearest neighbor. Here the age assigned the ith point, A_i, is assumed to directly depend on the age of the point in the sediment just below the ith point, A_{i-1}. Its dependence on the rest of the alignment variables stems from the fact that A_{i-1} is dependent on A_{i-2}, which in turn is dependent on A_{i-3} and so on back to the bottom of the record. This Markov chain dependence structure is a key assumption of HMM-Match and Match.

a) Remark 3.a: HMM-Match is said to be hidden because the alignment variables and the sedimentation event leading to them are not observed directly.
b) Remark 3.b: As we will see in the next component of this primer, the Markov character of HMM-Match greatly facilitates the computational requirements for draw inferences on these hidden variables. Specifically, the time complexity of an algorithm that explicitly accounts for the entire history grows exponentially with the numbers of data points in the input and the target, whereas the Markov assumption reduces the computational complexity to the product of the lengths of these two records.
c) Remark 3.c: The original Match algorithm also capitalizes on the assumption of Markov dependence as the basis of the dynamic programming algorithm that it employs.
d) Remark 3.d: The implementation employed in HMM-Match actually follows the paleoceanography convention of progressing back through time (down-core) rather than in forward time (up-core) as described in this primer. Since the transition matrix we use is symmetric this does not affect the results.

4) An important feature addressed by HMM-Match concerns sedimentation rate variability. Specifically, the probability of a particular sedimentation rate between i-1 and i, is assumed dependent upon the sedimentation rate between points i-2 and i-1. Returning to Figure 1, the sedimentation rate between i-1 and i is $\rho_i = \frac{d_i - d_{i-1}}{A_i - A_{i-1}}$, where d_i is the measured depth of the ith input point. HMM-Match, which makes inferences about times of deposition, employs inverse sedimentation rates, which are defined as $r_i = \frac{A_i - A_{i-1}}{d_i - d_{i-1}}$. HMM-Match uses the transition probabilities of a Markov chain, $\phi(r_{i-1}, r_i)$, to account the dependence of adjacent sedimentation rates. Since the sedimentation rate r_i's are functions of A_i's, they are also random variables. This relaxes the nearest neighbor assumption to one that allows dependence on the two nearest neighbors, which makes HMM-Match a second-order Markov model.

Transition probabilities give probability distributions for the sedimentation rates at point i, given the sedimentation rates at i-1. For example, in Figure 1b the probability of the transition from point A to point D models the dependence on which of the three transitions were taken to get to A from A', A'', or A'''. As described in the main text the probabilities of these transition rates, $\phi(r_{i-1}, r_i)$, were estimated from independent radiocarbon data.

Thus, although the probability of a given age assignment A_i for the ith input data point does pend on all others, this dependence can be fully accounted for by the age assignment of A_{i-1} and A_{i-2}, which provide information about whether point i-1 was aligned using a ratio of ~1:1, >1 (expansion) or <1 (contraction). This Markov property is specified mathematically as follows:

$$P\left(A_i \mid A_1, A_2, ..., A_{i-1}\right) = P\left(A_i \mid A_{i-1}, A_{i-2}\right), i = 3,4,...,n$$

a) Remark 4.1: So far this primer describes the manner in which HMM-Match models unknown sedimentation rates, but has not yet described how the proxy data are incorporated. From the prospective of Bayesian statistics, we have specified the prior model, but not yet the likelihood.

5) In HMM-Match the proxy data are incorporated using an emission model. The emission model accounts for the fact that even if we knew the correct alignment, of the ith input point to a specific point in the stack, say $A_i = j$, measurement errors and spatial variability are likely to result in differences between the two δ^{18}O values. To account for expected differences between the stack and an individual core, HMM-Match uses a probability distribution and associated random variables, $\mathbf{V} = \left(V_1, V_2, ..., V_n\right)$ to model potential observed δ^{18}O values corresponding to each point in the LR04 stack. The known measured δ^{18}O value at the ith point in the input is v_i. HMM-Match assumes that when $A_i = j$, the probability density for the event that observation $V_i = v_i$ follows a normal distribution with mean $w_j + \mu$ and variance σ^2, $N(w_j + \mu, \sigma^2 \mid A_i = j)$ in which w_j is the δ^{18}O value for the jth point in the stack,

σ^2 is the variance parameter to be estimated, and μ is an offset parameter that allows for potential shift in the mean $\delta^{18}O$ of each record from the mean $\delta^{18}O$ values in the stack (e.g., due to spatial variability in the ocean or interlaboratory calibration differences).

a) Remark 5.1: As noted in the main text we checked the residual of the model for agreement with the normality assumption.
b) Remark 5.2: In the jargon of HMMs we say that the proxy, $\delta^{18}O$, was emitted by the HMM.
c) Remark 5.3: Notice the model assumes that if the i^{th} input point is aligned to the j^{th} point in the stack, $A_i = j$, then probability of emitting the observed proxy, v_i, does not depend on any other variables in the model. This is another conditional independence assumption of the HMM-Match, also used by Match.

An important advantage of a generative model is its requirement to mathematically represent the underlying processes and the assumptions behind them. As we have described here, the HMM-Match generative model iteratively simulates the deposition of each sediment layer on top of the previous layer using a Markov assumption of conditional independence, while its emission model represents noise and spatial variations in $\delta^{18}O$. Careful attention to these assumptions is important because all the results of HMM-Match are a mathematical consequence of this generative model (and, of course, the data).

II. Alignment Inference Algorithm

1) As described above, the unknown events are modeled by HMM-Match in a forward direction from the sedimentation events (and the associated alignment with the stack) that led to the observed $\delta^{18}O$ values in the input. However, the logic of the dependence of the variables must be reversed to draw inferences about the alignment from the proxy data. This is done using Bayes rule. Because all the variables of the model are interrelated, the application of Bayes rule is not simple. Nevertheless, using the assumptions of conditional independence, Bayes rule can be employed to infer the

joint probability distribution of all unknown alignment variables using a two-stage algorithm: one that proceeds forward in time, up-core, and another that proceeds down-core. Both of these algorithms were derived directly from probability theory.

2) The first stage, which is known as the "forward" algorithm, works up-core from the lower right corner of Figure 1b step-by-step to the upper left corner of Figure 1b. This first stage, which sums probabilities over all possible alignments, uses the probability theory principle known as marginalization. Conditional independence allows the forward algorithm to step through the alignment matrix column-by-column. The key idea behind this column-by-column progression is that the algorithm solves a progressively growing set of sub-problems, each corresponding to the data from the deepest (oldest) δ^{18}O observation up to the ith observation $i = 1, 2, ..., n$ in the proxy sequence. Recall that in a Markov chain everything about the history of events is captured by the probability distribution of events that immediately precede it. Thus, when the algorithm fills in the probabilities for ith column of the alignment matrix it only uses the results stored in the (i-1)st and (i-2)nd columns, and not all the histories that lead to them. When the forward algorithm has completed summing probabilities to the upper left hand corner of the alignment matrix the last sub-problem solved is the full alignment problem (see below).

3) At this point Bayes rule can be employed to find the probability of any specified alignment, $\mathbf{A}^*$, as follows:

Posterior Emission Transition

$$P\left(\mathbf{A}^* \mid \mathbf{v}, \mathbf{w}, \mu, \sigma^2, \phi\left(r_i, r_{i-1}\right),\right) = \frac{P\left(\mathbf{v} \mid \mathbf{A}^*, \mu, \sigma^2, \mathbf{w}\right) P\left(\mathbf{A}^* \mid \phi\left(r_i, r_{i-1}\right)\right)}{\sum_{\mathbf{A} \in \Omega} P\left(\mathbf{v} \mid \mathbf{A}, \mu, \sigma^2, \mathbf{w}\right) P\left(\mathbf{A} \mid \phi\left(r_i, r_{i-1}\right)\right)}$$

, where $\mathbf{w} = \left(w_1, w_2, ..., w_m\right)$ is the vector of the δ^{18}O values in the stack, and $\nu = \left(\nu_1, \nu_2, ..., \nu_n\right)$ is the vector the δ^{18}O values in the input core. In Bayesian jargon the left hand side of this equation is call a posterior probability. The very large sum

over all possible alignments in the denominator is returned at the last step of the forward algorithm, and the value stored there is the denominator in Bayes rule. This result can be useful but it doesn't tell how to select the particular alignment $\mathbf{A}^*$. In computer science jargon the equation is not constructive.

4) Stochastic back trace algorithm: A procedure that selects examples of $\mathbf{A}^*$ that are exactly proportional to their probability according to Bayes rule is a useful way to select alignments because such a representative sample can yield unbiased estimates of nearly any aspect of this space. The stochastic back trace algorithm provides a means to obtain these samples. This algorithm begins at the top of the core (upper left corner in Figure 1b) and progressively samples alignments of input data points from the core-top down to the bottom of the input sequence. In so doing, the back trace algorithm captures information on the dependencies from the data points above it, while also using the stored matrix calculated by the forward stage to account for all the data points below it.

   In the first back step, the algorithm stochastically draws a value for the last alignment variable A_n. On the next step it calculates the probability of the transition from A_n to A_{n-1} and combines this with the information from the forward step at n-1. It then proceeds recursively back down the columns in Figure 1b. In so doing it samples a full alignment $\mathbf{A} = \left(A_1, A_2, ..., A_n\right)$. As probability theory shows, these samples are drawn in exact proportion to their probability because these two stages communicate information from both ends of the input record. The specifics of this inference algorithm are given in the supplement.

5) In this application we use 1000 stochastic back trace samples to obtain alignment confidence limits. Each sample alignment assigns every data point from the input to one age in the stack. Then, confidence limits are generated by identifying the range of stack ages that encompass 95% of the sampled alignments for each input point (i.e. from the 25th youngest age to the 975th oldest age). The tabulation of these confidence

limits for all points in the input yields estimated confidence bands for the full alignment.

a) Remark 5.1: These confidence limits quantify alignment uncertainty but not the uncertainty associated with age model of the alignment target.
b) Remark 5.2: An alternative backward summation algorithm can be employed in combination with the forward algorithm to obtain the intermediate results called marginal probabilities that can also be used to construct confidence intervals.

6) Often describing a sequence of unknowns by a probability distribution is not sufficient, because users want a specific prediction. Statistical decision theory provides a mean to return a "best" prediction, an estimate, once the meaning of best is well defined. In statistical decision theory best is defined using a loss function that specifies how the penalty increases as a function of the magnitude of difference between points in the space of all unknowns (here alignments) and an estimate. Most frequently the single most probable assignment of the unknowns is given for the "best" estimate. However, under this loss function there is no assurance that all of the components of this estimate will stay within the 95% confidence limits(3). An example of such a case is given in the supplement. Instead HMM-Match solves for the median alignment, which uses a loss function defined by the sum of the absolute differences between the estimator (i.e., median alignment) and the points in the space of all sampled alignments. By minimizing the expected values of these differences, the median estimator pulls the components of the estimate (i.e., each A_i) to the center of the distribution, and, thus, each estimated age in the median alignment stays within its 95% confidence limits.
   a) Remark 6.1: The generative model allows us to check for errors in the software code because we can specify values for unknown parameters, simulate the alignments, and generate sampled proxy values. Then, taking the generated proxy values as data, we can run HMM-Match while pretending that the parameters and the alignment variables are unknown. We check for coding errors by comparing the known simulated values to those inferred by the algorithm and verifying that they are within the tolerances of the model.

III. Parameter estimation

1) To this point in the primer we have assumed that all the parameters of the model are known. While most of them have been obtained from independent sources, the variance σ^2 and the offset parameter, μ, must be estimated using the data in each input record. This would be easy if we knew the alignment, but we don't. However, the Baum-Welch expectation maximization (EM) algorithm delivers maximum likelihood estimates of these parameters when the alignment is unknown. This algorithm is iterative, and like most nonlinear optimization algorithms, begins with a starting guess and iterates between an expectation E-step and a maximization M-step. Conceptually, on each iteration the E-step uses the probabilities of the space of alignments to find the expected values of the variables needed to find maximum likelihood estimates, known as the sufficient statistics, of the data that we would have if we knew the alignment; then the algorithm substitutes these expected values for the unknown values into its maximization step. The maximization M-step uses these expected values to update the parameter estimates and returns them for use in the next E-step. Theory shows that the likelihood will increase at each iteration(4); the algorithm stops when there is no further increase in the likelihood. Multiple initial guesses are employed to find the global optimum. In this application, few initial guesses are required because there are only two unknown being estimated. A more comprehensive description of the Baum-Welch algorithm is given by Rabiner (2) and Durbin et.al (1).

References

1. Durbin R, Eddy S, Krogh A, Mitchison G. Biological Sequence Analysis. Cambridge, UK: Cambridge University Press; 1998. 356 p.
2. Rabiner L. A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of The IEEE. 1989;77(2):257-86.
3. Carvalho LE, Lawrence CE. Centroid estimation in discrete high-dimensional spaces with applications in biology. Proceedings of the National Academy of Sciences. 2008;105(9):3209-14. doi: 10.1073/pnas.0712329105.

4. Dempster; AP, Laird NM, Rubin DB. Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society Series B. 1977;39(1):1-38.