IEOR 265 - Lecture 3
Inference and Estimation on HMM

1 Forward-Backward Algorithm

We now explore another algorithm that has some ties with the Viterbi Algorithm: The Forward-Backward Algorithm. This algorithm exploits the idea of recursion to provide a very powerful tool in statistical inference and it is particularly useful when applied in conjunction with Graphical Models [1]. Consider again the HMM model from the last lecture but now instead of the conditional probabilities of the observations $r(z|i,j)$ depending on both the states i and j, let's suppose it depends only on the initial state i, so $r(z|i) = \Pr(z_k = z, x_k = i), \forall k \in \{0, ..., N\}$. In addition, suppose for each hidden state x_k we collect one observation z_k. Then we can represent the HMM by the following figure:

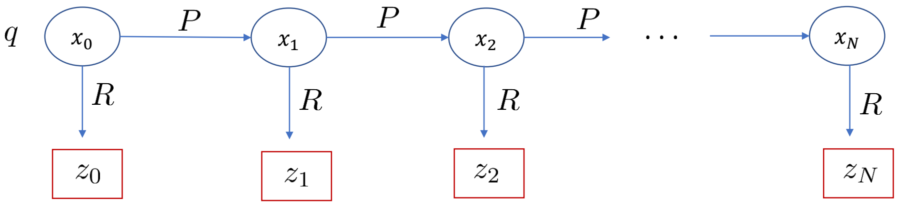

Figure 1: Schematic example of a HMM.

On the previous lecture, we were interested in obtained the most likely sequence of the hidden states $(\hat{x}_0, ..., \hat{x}_N)$ given the sequence of observations Z_N. It is simple to see how the Viterbi Algorithm extends to the new formulation above, and it is left as an exercise.

Now, we would like to obtain, instead of the most-likely sequence, the marginal distribution of some state x_k given our observations. This problem is actually sub-divided as follows:

1. Computing $\Pr(x_k|z_0, ..., z_k)$ for all possible values of x_k. This is called the *Filtering Problem.*

2. Computing $\Pr(x_k|z_0, ..., z_s), k > s$ for all possible values of x_k. This is called the *Prediction Problem.*

3. Computing $\Pr(x_k|z_0, ..., z_s), k < s$ for all possible values of x_k. This is called the *Smoothing Problem.*

These three problems are all related so let's focus on the smoothing problem where we have access to the full information sequence $Z_N = (z_0, ..., z_N)$ and we wish to compute the entire distribution $\Pr(x_k|Z_N)$ for some hidden state x_k. By applying Bayes' rule twice we obtain:

$$\Pr(x_k|Z_N) = \frac{\Pr(Z_N|x_k)\Pr(x_k)}{\Pr(Z_N)} \tag{1}$$

Now we use the fact that by conditioning on x_k the observations before and after stage k are independent:

$$\frac{\Pr(Z_N|x_k)\Pr(x_k)}{\Pr(Z_N)} = \frac{\Pr((z_0, ..., z_k)|x_k)\Pr((z_{k+1}, ..., z_N)|x_k)\Pr(x_k)}{\Pr(Z_N)} \tag{2}$$

Then we can regroup:

$$\frac{\Pr(Z_N|x_k)\Pr(x_k)}{\Pr(Z_N)} = \frac{\Pr(z_0, ..., z_k, x_k)\Pr((z_{k+1}, ..., z_N)|x_k)}{\Pr(Z_N)} \tag{3}$$

and we define the following quantities:

$$\alpha(x_k) = \Pr(z_0, ..., z_k, x_k),\ \beta(x_k) = \Pr(z_{k+1}, ..., z_N|x_k) \tag{4}$$

And we can write:

$$\frac{\Pr(Z_N|x_k)\Pr(x_k)}{\Pr(Z_N)} = \frac{\alpha(x_k)\beta(x_k)}{\Pr(Z_N)} \tag{5}$$

Note that both quantities have a nice intuitive meaning: $\alpha(x_k)$ is the probability of obtaining the observation sequence $(z_0, ..., z_k)$ and finishing the observations with the state x_k. This quantity evolves *forward* in time, as we shall see. On the hand, $\beta(x_k)$ represent the probability of observing the future sequence $(z_{k+1}, ..., z_N)$ given that we start from state x_k and it evolves *backward* in time.

Now let's turn our attention to the term $\Pr(Z_N)$: Notice that we can write the following equality:

$$\Pr(x_k|Z_N)\Pr(Z_N) = \alpha(x_k)\beta(x_k) \tag{6}$$

Now if we sum over all possible values of $x_k \in \mathcal{X}_k$ we obtain:

$$\Pr(Z_N) = \sum_{x \in \mathcal{X}_k} \alpha(x)\beta(x) \tag{7}$$

Lastly, be defining $\gamma(x_k) = \Pr(x_k|Z_N)$, we write compactly:

$$\gamma(x_k) = \frac{\alpha(x_k)\beta(x_k)}{\sum_{x \in \mathcal{X}_k} \alpha(x)\beta(x)} \tag{8}$$

Hence the smoothing problem is solved once we are able to compute $\alpha(x_k)$ and $\beta(x_k)$ for every possible value of x_k. This is done by a forward and backward recursion. First we write for $\alpha(x_{k+1})$:

$$\alpha(x_{k+1}) = \Pr(z_0, ..., z_{k+1}, x_{k+1}) = \Pr(z_0, ..., z_{k+1}|x_{k+1})\Pr(x_{k+1}) =$$

$$= \Pr(z_0, ..., z_k | x_{k+1}) \Pr(z_{k+1} | x_{k+1}) \Pr(x_{k+1}) =$$

$$= \Pr(z_0, ..., z_k, x_{k+1}) \Pr(z_{k+1} | x_{k+1}) =$$

$$= \sum_{x_k} \Pr(z_0, ..., z_k, x_k, x_{k+1}) \Pr(z_{k+1} | x_{k+1}) =$$

$$= \sum_{x_k} \Pr(z_0, ..., z_k, x_{k+1} | x_k) \Pr(x_k) \Pr(z_{k+1} | x_{k+1}) =$$

$$= \sum_{x_k} \Pr(z_0, ..., z_k | x_k) \Pr(x_{k+1} | x_k) \Pr(x_k) \Pr(z_{k+1} | x_{k+1}) =$$

$$= \sum_{x_k} \Pr(z_0, ..., z_k, x_k) \Pr(x_{k+1} | x_k) \Pr(z_{k+1} | x_{k+1}) =$$

$$= \sum_{x_k} \alpha(x_k) p_{x_k, x_{k+1}} r(z_{k+1} | x_{k+1}) \tag{9}$$

Note that this recursion is written forward in time: First we compute all the values of $\alpha(x_k)$ then we use them to compute all the values of $\alpha(x_{k+1})$. To start the forward recursion, we initialize for $k = 0$:

$$\alpha(x_0) = \Pr(z_0, x_0) = \Pr(z_0 | x_0) = r(z_0 | x_0) q(x_0) \tag{10}$$

In addition, we can "vectorize" the computation by writing the vectors and matrices:

$$\alpha_{k+1}^\top = \alpha_k^\top P \ R_{z_{k+1}, x_{k+1}} \tag{11}$$

where $R_{z_{k+1}, x_{k+1}}$ is a diagonal matrix where the $i'th$ diagonal element is equal to $r(z_{k+1} | x_{k+1} = i)$; P is the Markov Chain transition matrix; and α_k is a vector of length m, where each element i is equal to $\alpha(x_k)$ evaluated at $x_k = i$. This expression is useful to make explicit the fact that this recursion is $O(m^2)$, thus it can be done very efficiently. Since we have to the above for every stage, the overall complexity of the forward pass is $O(m^2 N)$.

The backward recursion to compute the $\beta(x_k)$ follows a similar derivation:

$$\beta(x_k) = \Pr(z_{k+1, ..., z_N | x_k}) =$$

$$= \sum_{x_{k+1}} \Pr(z_{k+1}, ..., z_N, x_{k+1} | x_k) =$$

$$= \sum_{x_{k+1}} \Pr(z_{k+1}, ..., z_N | x_{k+1}, x_k) \Pr(x_{k+1} | x_k) =$$

$$= \sum_{x_{k+1}} \Pr(z_{k+2}, ..., z_N | x_{k+1}) \Pr(z_{k+1} | x_{k+1}) \Pr(x_{k+1} | x_k) =$$

$$= \sum_{x_{k+1}} \beta(x_{k+1}) p_{x_k, x_{k+1}} r(z_{k+1} | x_{k+1}) \tag{12}$$

Now, to start this recursion we can't really use $\beta(x_N)$ since there is no observation y_{N+1}. This is solved by initializing $\beta(x_N) = 1, \forall x_N \in \{1, ..., m\}$. And we can verify this gives indeed the correct value for $\beta(x_{N-1})$:

$$\beta(x_{N-1}) = \Pr(z_N | x_{N-1}) = \sum_{x_N} \Pr(z_N, x_N | x_{N-1}) =$$

$$= \sum_{x_N} \Pr(z_N|x_N)\Pr(x_N|x_{N-1}) = \sum_{x_N} p_{x_{N-1},x_N} r(z_N|x_N) \tag{13}$$

Similarly we can write in vector form:

$$\beta_k = P\ R_{z_{k+1},x_{k+1}} \beta_{k+1} \tag{14}$$

where β_k is a vector of length m, where each element i is equal to $\beta(x_k)$ evaluated at $x_k = i$. The complexity in the computation is the same, so the overall complexity of both forward and backward to compute all necessary quantities for inference is $O(m^2N)$.

Lastly, note that we can compute $Pr(Z_N)$ by using:

$$\Pr(Z_N) = \sum_{x_N} \alpha(x_N)\beta(x_N) = \sum_{x_N} \alpha(x_N) \tag{15}$$

Hence, the probability of observing any sequence Z_N can be calculated after one forward pass. To obtain the posterior marginals, however, we need both passes, since:

$$\Pr(x_k|Z_N) = \frac{\alpha(x_k)\beta(x_k)}{\Pr(Z_N)} \tag{16}$$

If for example, we would like the so solve the Filtering problem, that is also readily available:

$$\Pr(x_k|z_0, ..., z_k) = \frac{\alpha(x_k)}{\Pr(Z_N)} \tag{17}$$

This fact is important for numerical stability issues: Note that the recursions involves **sums of product** of probabilities. This can quickly lead to underflow problems (i.e.: "numbers" getting very small). In order to solve that, on can use appropriate normalizations to ensure that the $\alpha(x_k)$ are well-scaled.

Another practical consideration gives birth to a variation of the algorithm in which the $\beta(x_k)$ quantities are not needed and we can "throw away" the observations z_k as we obtain them in the forward pass.

The variation begins the same: We perform the forward pass to compute $\alpha(x_k)$. But in the backward pass we explicitly compute the $\gamma(x_k) = \Pr(x_k|z_0, ..., z_N)$ quantities (Eq.8). Namely we can start by writing:

$$\Pr(x_k|x_{k+1}, z_0, ..., z_N) = \Pr(x_k|x_{k+1}, z_0, ..., z_t) \tag{18}$$

Now, if we multiply both sides by $\Pr(x_{k+1}|z_0, ..., z_N)$, and then sum over x_{k+1}, we get:

$$\gamma(x_k) = \sum_{x_{k+1}} \Pr(x_k|x_{k+1}, z_0, ..., z_N)\Pr(x_{k+1}|z_0, ..., z_N) =$$

$$= \sum_{x_{k+1}} \Pr(x_k|x_{k+1}, z_0, ..., z_t)\Pr(x_{k+1}|z_0, ..., z_N) =$$

$$= \sum_{x_{k+1}} \frac{\Pr(x_k, x_{k+1}, z_0, ..., z_t)}{\sum_{x_k} \Pr(x_k, x_{k+1}, z_0, ..., z_t)} \Pr(x_{k+1}|z_0, ..., z_N) =$$

$$= \sum_{x_{k+1}} \frac{\Pr(x_k, z_0, ..., z_t)\Pr(x_{k+1}|x_k)}{\sum_{x_k} \Pr(x_k, z_0, ..., z_t)\Pr(x_{k+1}|x_k)} \Pr(x_{k+1}|z_0, ..., z_N) =$$

$$= \sum_{x_{k+1}} \frac{\alpha(x_k) p_{x_k, x_{k-1}}}{\sum_{x_k} \alpha(x_k) p_{x_k, x_{k-1}}} \gamma(x_{k+1}) \tag{19}$$

And the recursion is initialized with $\gamma(x_N) = \alpha(x_N)$. Note that this recursion does not use neither the β variables and the observation data z_k. So it can be done in a "real-time" fashion as we do not need to keep record of past observations.

2 Estimation in HMM: Full Information Case

A key part of the our development with HMM's lies in data that was necessary to carry out the Viterbi Algorithm and the Forward-Backward Algorithm: Namely, the transition matrix P, the initial state distribution $q(\cdot)$, and the conditional observation probabilities $r(\cdot|x)$. On this section we will explore two ways of estimating those parameters: (1) first we will assume we have a data set of transitions and observations available; and (2) where we will assume we have a data set only of observations (which would be the case in a practical application, as the system states are hidden).

As always, we will frame the estimation problem as an optimization problem where the decision variables are now related to P, q, and r, instead of the hidden states variables. In order to make the presentation simple, we will show the estimation methods for the case where the observations $z_k \sim \Pr(\cdot|x_k)$ follow a multinomial distribution. Namely as follows:

1. We let the hidden state x_k be a vector with M components, and if the underlying markov chain is in state i at stage k, then we set the $i'th$ element of x_k to be one and all the other elements to be zero. (An intuitive way to think about that is by imagining a dice with M faces and we record which face it lands on with 1 or a "tick").

2. We let the observation z_k be a vector with M components, such that only a single component j is equal to one and all the other elements are equal to zero. We let $r_{i,j}$ to denote the conditional probability that the $j'th$ component of z_k is equal to one, **given** that the $i'th$ component of x_k is one, that is: $r_{i,j} = \Pr(z_k^{(j)} = 1 | x_k^{(i)} = 1)$.

3. Hence in our estimation problem, our decision variables are the matrix P, where $p_{i,j}$ is the probability for the markov chain to transition from i to j; The conditional probability matrix R, where r_i, j is defined as the previous item; the vector q, where q_i is the probability of the markov chain to start at stage i.

First, let's suppose we acquired (say, by simulation) a data set of transitions $X = \{(x_0^l, ..., x_N^l)\}_{l=1}^L$, where we repeat the simulation L times, and we let the superscript denote each experiment $l \in \{1, ..., L\}$. In addition, for each experiment we also collect a sequence of observations $Z = \{(z_0^l, ..., z_N^l)\}_{l=1}^L$. Furthermore assume that each experiment are conducted in an *i.i.d.* manner.

The typical way of estimating our parameters is via *Maximum Likelihood Estimation (MLE)*: We would like to find the set of parameters $\theta = (A, R, q)$

that maximize the probability of observing our collected data set (X, Z). With that in mind we can write the joint probability as follows:

$$\Pr(X, Z|\theta) = \prod_{l=1}^{L} \Pr(x_0^l, ..., x_N^l, z_0^l, ..., z_N^l|\theta) = \tag{20}$$

since each experiment l is i.i.d. and we highlight the the above probability is given the parameters θ. Now we condition on the initial state and apply the Markov property, in a similar way as before (we refer to the Lecture 2 for a detailed step-by-step on how to obtain Eq.21):

$$\Pr(X, Z|\theta) = \prod_{l=1}^{L} q(x_0^{(l)}) \prod_{k=1}^{N-1} (p_{x_k^{(l)}, x_{k+1}^{(l)}}) \prod_{k=1}^{N-1} (r(z_k^{(l)}|x_k^{(l)}) \tag{21}$$

Now we will introduce the idea of counts: We let $m_{i,j}$ be the number of times a transition from i to j is "seen" in the data (recall that our data now are the state and the observation sequences for the L experiments). Similarly, let $\eta_{i,j}$ be the number of times we "see" the observation state to be j when the underlying markov chain state is i. Lastly, let v_i the number of times the experiments begin at state i. Based on our definitions for x and z and the counts defined above, we can see that (we leave the verification as an exercise):

$$m_{i,j} = \sum_{l=1}^{L} \sum_{k=0}^{N-1} x_k^{l,i} x_{k+1}^{l,j} \tag{22}$$

$$\eta_{i,j} = \sum_{l=1}^{L} \sum_{k=0}^{N-1} x_k^{l,i} z_k^{l,j} \tag{23}$$

$$v_i = \sum_{l=1}^{L} x_0^{l,i} \tag{24}$$

Now, using the definition of our decision variables, we can make substitutions in Eq.21 to obtain:

$$\Pr(X, Z|\theta) = \prod_{l=1}^{L} \prod_{i=1}^{p} [q_i]^{x_0^{l,i}} \prod_{k=1}^{N-1} \prod_{i=1}^{M} \prod_{j=1}^{M} [p_{i,j}]^{x_k^{l,i} x_{k+1}^{l,j}} \prod_{k=1}^{N-1} \prod_{i=1}^{M} \prod_{j=1}^{M} [r_{i,j}]^{x_k^{l,i} z_k^{l,j}} \tag{25}$$

Now, taking the log Eq.25, we can write:

$$\ln(\Pr(X, Z|\theta)) =$$
$$\sum_{l=1}^{L} \left(\sum_{i=1}^{p} x_0^{l,i} \ln(q_i) + \sum_{k=1}^{N-1} \sum_{i=1}^{M} \sum_{j=1}^{M} x_k^{l,i} x_{k+1}^{l,j} \ln(p_{i,j}) + \sum_{k=1}^{N-1} \sum_{i=1}^{M} \sum_{j=1}^{M} x_k^{l,i} z_k^{l,j} \ln(r_{i,j}) \right) \tag{26}$$

Now, rearranging the summations and using our counts (Eq.22-24), we can write:

$$\ln(\Pr(X, Z|\theta)) = \sum_{i=1}^{p} v_i \ln(q_i) + \sum_{i=1}^{M} \sum_{j=1}^{M} m_{i,j} \ln(p_{i,j}) + \sum_{i=1}^{M} \sum_{j=1}^{M} \eta_{i,j} \ln(r_{i,j}) \tag{27}$$

We take the opportunity to highlight a very important phenomenon that is showcased by Eq.27: Notice that by our manipulations we wrote the desired log-probability as a function of our counts m, η, and v. At the same time, the data (X, Z) completely "vanished" from the computation. More formally, the counts m, η, and v are the **sufficient statistics** for our estimation problem, that is, they describe all the information necessary to carry out the estimation. The concept of sufficient statistics is very important in high-dimensional settings, where we can fore-go using the entire data set (X, Z) (which can contain millions of data points), and only use the sufficient statistics (much smaller in comparison) to perform the estimation.

Now we are finally able to write the optimization problem related to the MLE:

$$\max_{P,R,q} \sum_{i=1}^{p} v_i \ln(q_i) + \sum_{i=1}^{M}\sum_{j=1}^{M} m_{i,j} \ln(p_{i,j}) + \sum_{i=1}^{M}\sum_{j=1}^{M} \eta_{i,j} \ln(r_{i,j}) \tag{28}$$

$$\text{s.t.: } \sum_{i=1}^{M} q_i = 1 \tag{29}$$

$$\sum_{j=1}^{M} p_{i,j} = 1 \tag{30}$$

$$\sum_{j=1}^{M} r_{i,j} = 1 \tag{31}$$

$$p_{i,j} \geq 0, r_{i,j} \geq 0, q_i \geq 0, \forall i, j \in \{1, ..., M\} \tag{32}$$

where we note that since our decision variables are probabilities, we need to enforce that they sum up to one and are nonnegative. A simple way of solving the above problem, is by "ignoring" the nonnegativities (Eq.32), and then solve the equality constraint problem via the Lagrange Multipliers Method. Luckily, the solution that comes out of this will be nonnegative, hence it will be the optimal solution as well, since the objective function is concave. We provide the optimal solution below and leave the actual derivation as an exercise:

$$p^*_{i,j} = \frac{m_{i,j}}{\sum_{k=1}^{M} m_{i,k}} \tag{33}$$

$$r^*_{i,j} = \frac{\eta_{i,j}}{\sum_{k=1}^{M} \eta_{i,k}} \tag{34}$$

$$q^*_i = \frac{v_i}{\sum_{k=1}^{M} v_k} \tag{35}$$

and one can verify that they are indeed nonnegative and sum to one.

3 Partial Information Case: The EM Algorithm

Now we will focus on the more difficult case, and the one that is often the case in practice: where we can only collect data from the observation variables z_k. Without observing the true system states, our objective function in the estimation suffers a slight but very crucial change. Let's begin by writing the probability that we obtained on the previous section:

$$l_c(Z;\theta) = \ln(\Pr(X,Z|\theta)) = \sum_{i=1}^{p} v_i \ln(q_i) + \sum_{i=1}^{M}\sum_{j=1}^{M} m_{i,j} \ln(p_{i,j}) + \sum_{i=1}^{M}\sum_{j=1}^{M} \eta_{i,j} \ln(r_{i,j}) \tag{36}$$

This expression assumes complete information (both the hidden states x and the observations z). Due to this, log-probability if often called the *complete log-likelihood* of the HMM, which we denote by $l_c Z;\theta$ where the subscript c stands for "complete". Now on the partial information case what we want instead is the following probability:

$$l(Z;\theta) = \ln(\Pr(Z|\theta)) = \sum_{l=1}^{L} \ln\left(\sum_{(x_0,\ldots,x_N)} \Pr(x_0,...,x_N,z_0^l,...,z_N^l|\theta)\right) \tag{37}$$

where the summation inside the logarithm is over all possible sequences of the underlying markov chain: We essentially performed a marginalization over all possible sequences of the underlying markov chain. Note how difficult Eq.37 is: we have a logarithm of a sum, instead of a sum of logarithms! This makes the problem much more challenging to analyze: Namely, observe that the above function **is not** concave! Eq.37 is often called the *incomplete log-likelihood* (since we are "missing" the data from the x_k variables).

Note that since we cannot observe the data set X, Eq.36 is essentially defining a random quantity. Let's assume for the moment that we have in our hands a conditional distribution $q(x_0,...,x_N|z_0,...,z_N)$ and we use it to "average out" the random quantity in Eq.36. Namely, we can define the *expected complete log-likelihood*:

$$\mathbb{E}_q\left[l_c(Z;\theta)\right] = \sum_{l=1}^{L}\sum_{(x_0,\ldots,x_N)} q(x_0,...,x_N|z_0^l,...,z_N^l)\ln\left(\Pr(x_0,...,x_N,z_0^l,...,z_N^l|\theta)\right) \tag{38}$$

We note that the above quantity is **not** random: it is a deterministic function of Z (our data set) and θ (the parameters). The intuition behind this approach is as follows: If we are smart and we choose the distribution q "well" the expected complete log-likelihood (Eq.38) may be very similar to the log-likelihood of interest (Eq. 37). This is the idea behind *Surrogate Optimization*, where we "replace" a hard objective function for a simple one and hope that our approximation is close-enough to the true function: By maximizing the surrogate function (Eq.38) we may find a value of θ that is actually an improvement for the true function (Eq.37).

This is the basic idea one of the most important (and used) algorithms in Machine Learning and Statistical Inference: **The EM Algorithm**, where "EM" refers to expectation-maximization.

Let's start with the derivation. First we show that for any conditional distribution $q(x|z) = q(x_0, ..., x_N|z_0, ..., z_N)$ we can provide a lower bound on the true log-likelihood:

$$l(Z;\theta) = \ln(\Pr(Z|\theta)) = \sum_{l=1}^{L} \ln\left(\sum_{x} \Pr(x, z^l|\theta)\right) = \tag{39}$$

$$\sum_{l=1}^{L} \ln\left(\sum_{x} \frac{q(x|z)}{q(x|z)} \Pr(x, z^l|\theta)\right) \geq \tag{40}$$

$$\sum_{l=1}^{L} \sum_{x} q(x|z) \ln\left(\frac{\Pr(x, z^l|\theta)}{q(x|z)}\right) = \mathcal{L}(Z; q, \theta) \tag{41}$$

where we $x = (x_0, ..., x_N)$ and $z^l = (z_0^l, ..., z_N^l)$ in order to simplify notation and on the last line we defined the *auxiliary function* $\mathcal{L}(Z; q, \theta)$. To achieve this result we used Jensen Inequality on Eq.40-41 coupled with the concavity of the logarithm function. So what this shows is that we have:

$$l(Z;\theta) \geq \mathcal{L}(Z; q, \theta), \, \forall\theta \tag{42}$$

Now the EM algorithm is essentially an application of the **Coordinate Ascent Method** in the auxiliary function $\mathcal{L}(Z; q, \theta)$: Namely at iteration $t+1$ we first maximize $\mathcal{L}(Z; q, \theta^{(t)})$ with respect to q (for a fixed $\theta^{(t)}$), thus obtaining a conditional distribution $q^{(t+1)}$; Then we maximize $\mathcal{L}(Z; q^{(t+1)}, \theta)$ with respect to θ (for a fixed $q^{(t+1)}$), thus obtaining an updated parameters values $\theta^{(t+1)}$. The EM algorithm iterates between these two steps repeatedly. We state them precisely below, with their namesake names:

$$\textbf{E-step:} \qquad q^{(t+1)} = \arg\max_{q} \mathcal{L}(Z; q, \theta^{(t)}) \tag{43}$$

$$\textbf{M-step:} \qquad \theta^{(t+1)} = \arg\max_{\theta} \mathcal{L}(Z; q^{(t+1)}, \theta) \tag{44}$$

We will explain soon why the E-step is called the expectation step. First we will show why maximizing the lower bound $\mathcal{L}(Z; q, \theta)$ is a good idea and how it connects with the complete log-likelihood. First we show that performing the M-step is equivalent as maximizing the expected complete log-likelihood. To see this, we can break $\mathcal{L}(Z; q, \theta)$ in two terms:

$$\mathcal{L}(Z; q, \theta) = \sum_{l=1}^{L} \sum_{x} q(x|z) \ln\left(\frac{\Pr(x, z^l|\theta)}{q(x|z)}\right) =$$

$$= \sum_{l=1}^{L} \sum_{x} q(x|z) \ln\big(\Pr(x, z^l|\theta)\big) - \sum_{l=1}^{L} \sum_{x} q(x|z) \ln\big(q(x|z)\big) =$$

$$= \mathbb{E}_q\big[l_c(Z;\theta)\big] - \sum_{l=1}^{L} \sum_{x} q(x|z) \ln\big(q(x|z)\big) \tag{45}$$

and we can observe that the second term at the end of Eq.45 does not depend on θ, so maximizing the expected complete log-likelihood is the same as the M-step defined on Eq.44.

Now let's focus on the E-step, which as defined by Eq.43 is a maximization of the lower bound $\mathcal{L}(Z;q,\theta^{(t)})$ with respect to the conditional distribution $q(x|z)$. We claim that the optimal solution of the E-step is **always** given by the conditional probability $q^*(x|z) = \Pr(x|z,\theta^{(t)})$. To see observe that if we let $q^*(x|z) = \Pr(x|z,\theta^{(t)})$ and substitute on the definition of $\mathcal{L}(Z;q,\theta)$, we get:

$$\mathcal{L}(Z;\Pr(x|z,\theta^{(t)}),\theta^{(t)}) = \sum_{l=1}^{L}\sum_{x}\Pr(x|z,\theta^{(t)})\ln\left(\frac{\Pr(x,z^l|\theta)}{\Pr(x|z,\theta^{(t)})}\right) = \tag{46}$$

$$= \sum_{l=1}^{L}\sum_{x}\Pr(x|z,\theta^{(t)})\ln\left(\Pr(z|\theta^{(t)})\right) = \ln(\Pr(Z|\theta^{(t)})) = l(Z;\theta^{(t)}) \tag{47}$$

hence given that $l(Z;\theta)$ is an upper bound for $\mathcal{L}(Z;q,\theta^{(t)})$, this show that by setting $q(z|x)$ equal to $\Pr(x|z,\theta^{(t)})$, we match the bound, thus obtaining the optimal solution to the E-step. Therefore, the reason E-step is called the expectation step is because we only need to compute the conditional probabilities $\Pr(x|z,\theta^{(t)})$ and formulate the expected complete log-likelihood with these conditional probabilities. *That* is the expectation performed in the E-step:

$$\mathbb{E}_p\left[l_c(Z;,\theta^{(t)})\right] = \sum_{l=1}^{L}\sum_{(x_0,\ldots,x_N)}\Pr(x|z,\theta^{(t)})\ln\left(\Pr(x,z^l|,\theta^{(t)})\right) \tag{48}$$

And then the M-step improves the above expression by updating $\theta^{(t)}$ to $\theta^{(t+1)}$.

We want to stop for a moment to discuss how beautiful Eq.48 is: We can view this entire algorithm as a method that computes and updates **beliefs**: faced with the unknown nature of the world, that is the hidden states x, the best we can do is to try to formulate a belief, that is the distribution $q(x|z)$ of what the world is given our current observation z. Well, the best belief we can formulate if our goal is to "explain" the phenomenon that we are observing (the likelihood) is to precisely compute the conditional probabilities $(\Pr(x|z,\theta^{(t)}))$ and let that be our belief! Then we use those beliefs to find a "better" explanation for the world, namely the M-step, updating the underlying parameters. With these new parameters at hand, we then proceed to update our beliefs, namely the E-step. And we keep doing this until we are no longer able to improve.

This idea is so intuitive and appealing that it goes beyond Machine Learning and Statistical Inference, but finds ground in Game Theory, Economics, Psychology and many more areas! The intrinsic notion that the best we can do to explain nature given our (imperfect) observations is to formulate beliefs that match the conditional probabilities of the phenomenon occurring given what we know.

Let's now return to the problem at hand and conclude by showing that the EM algorithm, by maximizing the lower bound $\mathcal{L}(Z;q,\theta)$, can indeed maximize the true log-likelihood. Note that by the M-step we are increasing the lower bound $\mathcal{L}(Z;q,\theta)$, but that does not necessarily translates to increasing the true log-likelihood, since if there is a non-zero gap between, then the true log-likelihood may not increase even if the lower bound does. However, this gap is closed by the E-step, since by Eq.47 we have:

$$l(Z;\theta^{(t)}) = \mathcal{L}(Z;q^{(t+1)},\theta^{(t)}) \tag{49}$$

thus by making an increase in $\mathcal{L}(Z; q^{(t+1)}, \theta^{(t)})$ we also increase $l(Z;\theta)$. Hence the EM Algorithm continuously increase the value of $l(Z;\theta)$, which is what we wish to maximize.

3.1 Applying the EM algorithm to HMM

Now that the EM algorithm is properly derived, let's apply to our HMM setting. Recall that, in this case $\theta = (P, R, q)$. Let's first write down (again) the complete log-likelihood:

$$l_c(Z;\theta) = \ln(\Pr(X, Z|\theta)) = \sum_{i=1}^{p} v_i \ln(q_i) + \sum_{i=1}^{M}\sum_{j=1}^{M} m_{i,j} \ln(p_{i,j}) + \sum_{i=1}^{M}\sum_{j=1}^{M} \eta_{i,j} \ln(r_{i,j}) \tag{50}$$

and now the true log-likelihood:

$$l(Z;\theta) = \ln(\Pr(Z|\theta)) =$$

$$\sum_{l=1}^{L} \ln\Bigg(\sum_{(x_0,\ldots,x_N)} \prod_{i=1}^{p} [q_i]^{x_0^{l,i}} \prod_{k=1}^{N-1}\prod_{i=1}^{M}\prod_{j=1}^{M} [p_{i,j}]^{x_k^{l,i} x_{k+1}^{l,j}} \prod_{k=1}^{N-1}\prod_{i=1}^{M}\prod_{j=1}^{M} [r_{i,j}]^{x_k^{l,i} z_k^{l,j}} \Bigg) \tag{51}$$

Now we reference the solution for the full information case (Eq.33-35) and we recall that the E-step lies in computing the conditional expectations. In the HMM case, we replaced the hidden states x_k by their sufficient statistics, that is the counts m, η, v. Now, These counts are random since we do not observe X, so we compute their conditional expectations given the Z:

$$\mathbb{E}_p\big[\eta_{i,j}|Z;\theta^{(t)}\big] = \sum_{l=1}^{L}\sum_{k=0}^{N} \mathbb{E}\big[x_k^i|z^l, \theta^{(t)}\big] z_k^{j,l} =$$

$$= \sum_{l=1}^{L}\sum_{k=0}^{N} \Pr(x_k^i = 1|z^l, \theta^{(t)}) z_k^{j,l} = \sum_{l=1}^{L}\sum_{k=0}^{N} \gamma_k^{l,i} z_k^{l,j} \tag{52}$$

where $\gamma_k^{l,i}$ is equal to the $\gamma^l(x_k)$ evaluated at x_k such that $x_k^i = 1$. Note that $\gamma^l(x_k)$ are exactly what comes out of Forward-Backward Algorithm: the factors $\gamma^l(x_k)$ can be computed by the Forward-Backward Algorithm using $\theta^{(t)}$ as the underlying parameters and $(z_0^l, \ldots, z_N^l)$ as the observation sequence. We state the formula here for completeness:

$$\gamma^l(x_k) = \sum_{x_{k+1}} \frac{\alpha^l(x_k) p_{x_k, x_{k-1}}}{\sum_{x_k} \alpha^l(x_k) p_{x_k, x_{k-1}}} \gamma^l(x_{k+1}) \tag{53}$$

Next we write:

$$\mathbb{E}_p\big[m_{i,j}|Z;\theta^{(t)}\big] = \sum_{l=1}^{L}\sum_{k=0}^{N} \mathbb{E}\big[x_k^i x_{k+1}^i|z^l, \theta^{(t)}\big] =$$

$$= \sum_{l=1}^{L}\sum_{k=0}^{N} \Pr(x_k^i = 1, x_{k+1}^i = 1|z^l, \theta^{(t)}) = \sum_{l=1}^{L}\sum_{k=0}^{N} \psi_{k,k+1}^{l,i,j} \tag{54}$$

where $\psi^{l,i,j}_{k,k+1}$ denote $\psi^l(x_k, x_{k+1})$ for (x_k, x_{k+1}) such that $x^i_k = 1$ and $x^i_{k+1} = 1$. Where the quantities $\psi^l(x_k, x_{k+1})$ are derivatives of the $\gamma^l(x_k)(x_k)$ via the following formula:

$$\psi^l(x_k, x_{k+1}) = \Pr(x_k, x_{k+1}|z^l_0, ..., z^l_N) = \frac{\Pr(z_0, ..., z_N|x_k, x_{k+1})\Pr(x_k)}{\Pr(z_0, ...z_N)} =$$

$$= \frac{\Pr(z^l_0, ..., z^l_k|x_k)\Pr(z^l_{k+1}|x_{k+1})\Pr(z^l_{k+2}, ..., z^l_N|x_{k+1})\Pr(x_{k+1}|x_k)\Pr(x_k)}{\Pr(z^l_0, ...z^l_N)} =$$

$$= \frac{\alpha^l(x_k)\beta^l(x_{k+1})p_{x_k,p_{k+1}}r(z^l_{k+1}|x_{k+1})}{\Pr(z^l_0, ...z^l_N)} \tag{55}$$

which can be converted as function of the $\alpha^l(x_k)$ and $\gamma^l(x_k)$ (we leave the transformation as an exercise):

$$\psi^l(x_k, x_{k+1}) = \frac{\alpha^l(x_k)\gamma^l(x_{k+1})p_{x_k,p_{k+1}}r(z^l_{k+1}|x_{k+1})}{\alpha^l(x_{k+1})} \tag{56}$$

which can be computed directly via the Forward-Backward Algorithm using $\theta^{(t)}$ as the HMM parameters and $(z^l_0, ..., z^l_N)$ as the observation sequence. Lastly we can compute:

$$\mathbb{E}_p\left[v_i|Z;\theta^{(t)}\right] = \sum_{l=1}^{L} \mathbb{E}\left[x^i_0|z^l, \theta^{(t)}\right] = \sum_{l=1}^{L} \Pr(x^i_0 = 1, |z^l, \theta^{(t)}) = \sum_{l=1}^{L} \gamma^{l.i}_0 z^{l,i}_0 \tag{57}$$

With the estimated sufficient statistics in hand, we can write the M-step of the EM Algorithm, which for the HMM case is famous and it has a name, **The Baum-Welch Updates**:

$$r^{(t+1)}_{i,j} = \frac{\sum_{l=1}^{L}\sum_{k=0}^{N}\gamma^{l,i}_k z^{l,j}_k}{\sum_{l=1}^{L}\sum_{k=0}^{N}\sum_{u=0}^{M}\gamma^{l,i}_k z^{l,u}_k} = \frac{\sum_{l=1}^{L}\sum_{k=0}^{N}\gamma^{l,i}_k z^{l,j}_k}{\sum_{l=1}^{L}\sum_{k=0}^{N}\gamma^{l,i}_k} \tag{58}$$

where we used the fact that $\sum_{j=1}^{M} z^{l,j}_k = 1$. And:

$$p^{(t+1)}_{i,j} = \frac{\sum_{l=1}^{L}\sum_{k=0}^{N}\psi^{l,i,j}_{k,k+1}}{\sum_{l=1}^{L}\sum_{k=0}^{N}\sum_{u=0}^{M}\psi^{l,i,u}_{k,k+1}} = \frac{\sum_{l=1}^{L}\sum_{k=0}^{N}\psi^{l,i,j}_{k,k+1}}{\sum_{l=1}^{L}\sum_{k=0}^{N}\gamma^{l,i}_k} \tag{59}$$

where we used the fact that $\sum_{j=1}^{M}\psi^{i,j,l}_{k,k+1} = \gamma^{l,i}_k$. And lastly:

$$q^{(t+1)}_i = \frac{\sum_{l=1}^{L}\gamma^{l,i}_0 z^{l,i}_0}{\sum_{l=1}^{L}\sum_{j=1}^{M}\gamma^{l,j}_0 z^{l,j}_0} = \frac{\sum_{l=1}^{L}\gamma^{l,i}_0}{L} \tag{60}$$

where we used the fact that $z^{l,i}_0 = 1$ if and only if $\gamma^{l,i}_0 \neq 0$ and the fact that $\sum_{l=1}^{L}\sum_{j=1}^{M}\gamma^{l,j}_0 z^{l,j}_0 = L$, the number of experiments.

Hence the EM algorithm iterates between the Baum-Welch updates in the M-step (Eq.58-60) and the Forward-Backward Algorithm of the E-step (Eq.52-57), which is very efficient, even for high-dimensional settings, which showcases how powerful and useful this method can be in practical applications.

References

[1] M. J. Wainwright, M. I. Jordan *et al.*, "Graphical models, exponential families, and variational inference," *Foundations and Trends® in Machine Learning*, vol. 1, no. 1–2, pp. 1–305, 2008.