EECS E6720 Bayesian Models for Machine Learning

Columbia University, Fall 2016

Lecture 11, 12/1/2016

Instructor: John Paisley

- We next look at a model for sequential data. In this case, we assume we have a single sequence $(x_1, \dots, x_T)$ where each $x_t \in \{1, \dots, V\}$. We want to model the sequential information in this data.

- For simplicity, we will assume we have only one sequence for now. However, in reality we often have many sequences. I'll address the simple modifications that can be made to account for this at the end. For now, assume T is very large so we can have plenty of sequential information in this single sequence.

- Often the data is in $\mathbb{R}^d$. That would require a slightly different modeling approach than what we will discuss below. Incidentally, discrete-valued sequences often arise from what are originally continuous-valued sequences. For example, you can take the original data in $\mathbb{R}^d$ and quantize it by first running a clustering algorithm (K-means is the usual one) and then mapping $x_t \in \mathbb{R}^d$ to a value $x_t \in \{1, \dots, V\}$, where the value indicates which of the V clusters the original x_t was mapped to.

- Therefore, the following discussing on discrete sequences is useful for many different types of sequential data that aren't initially discrete-valued.

- We will next discuss a model for $x = (x_1, \dots, x_T)$ called a (discrete) hidden Markov model.

Hidden Markov model (HMM)

- First the high-level description. We associate each observation x_t in the sequence x with a hidden "state" s_t. Therefore, there is a sequence $s = (s_1, \dots, s_T)$ that corresponds to x where each element in s gives the state for the corresponding element in x. For now, the "state" is a totally abstract thing, much like the "cluster" was an abstract concept in the GMM.

- Often one can hope in advance to learn states that correspond to something meaningful (e.g., machine is "working" or "not working"), but for this lecture take a state to be something that is an abstract mathematical concept.

- There are three variables that define a K-state discrete HMM:

1. π : A K-dimensional probability vector used to generate s_1
2. A : A $K \times K$ Markov transition matrix. A_{ij} is the probability of transitioning to state j given that the current state is i
3. B : A $K \times V$ emission matrix. B_{iv} is the probability of observing $x = v$ given its corresponding state $s = i$.

- Model: Using these model variables, we generate data from the HMM as follows. Again, we have a sequence of data $x = (x_1, \ldots, x_T)$ where $x_t \in \{1, \ldots, V\}$. A K-state discrete HMM models this sequence as follows.
    - Generate state sequence: $s_1 \sim \text{Discrete}(\pi), \quad s_t | s_{t-1} \sim \text{Discrete}(A_{s_{t-1},:})$
    - Generate observation sequence: $x_t | s_t \sim \text{Discrete}(B_{s_t,:})$

- Each row in A is a probability distribution that decides what the next state will be. Therefore, $s_{t-1} \in \{1, \ldots, K\}$ simply picks out the correct row of A to use, which is what the notation is indicating. Similarly with B, the current state s_t picks out the probability distribution (i.e., row of B) to use to generate x_t.

- In this sense, a state corresponds to a probability distribution on an observation. It's worth thinking about how this relates to the mixture model. Each row of B is a data-level distribution much like a Gaussian was for the GMM. Each row of A is like a mixing distribution on which "cluster" (or "state" here) to pick to generate an observation. The HMM is like a mixture model where the "clusters" are not changing, but the *distribution* on the clusters is changing according to a first-order Markov process.

Maximum likelihood Expectation-Maximization (ML-EM)

- In most of what follows, our goal is to find a point estimate of $H = \{\pi, A, B\}$ that maximizes the marginal likelihood

$$\ln p(x|H) = \ln \sum_s p(x, s|H) \tag{1}$$

- Comments:
    1. From now on, we'll use H for π, A, B when they are all conditioned on.
    2. The sum is over all T-length sequences s. Since there are K^T possible sequences, this sum can't be directly done.
    3. However, with EM we never actually need to calculate the marginal explicitly. We only need to be able to write out the joint likelihood over x and s.
    4. We mention that the sum can be performed in a clever way using outputs from either the "forward" or the "backward" algorithms, both of which we will discuss later.

- Joint likelihood: As usual, we have to start here before we can do anything else.

$$\begin{aligned} p(x,s|H) &= p(x|s,B)p(s|A,\pi) \qquad (2) \\ &= \left[\prod_{t=1}^{T} p(x_t|s_t,B)\right]\left[p(s_1|\pi)\prod_{t=2}^{T} p(s_t|s_{t-1},A)\right] \\ &= \left[\prod_{t=1}^{T}\prod_{k=1}^{K}\prod_{v=1}^{V} B_{kv}^{\mathbb{1}(s_t=k)\mathbb{1}(x_t=v)}\right]\left[\prod_{k=1}^{K}\pi_k^{\mathbb{1}(s_1=k)}\right]\left[\prod_{t=2}^{T}\prod_{i=1}^{K}\prod_{j=1}^{K} A_{ij}^{\mathbb{1}(s_{t-1}=i,s_t=j)}\right] \end{aligned}$$

- Comments:

  1. $p(x|s,B)$: Given the state sequence s, x only depends on B. Also, if I know which state each observation came from, then the observations are all independent of each other. This is a result of the model definition. So we can write this likelihood as a product.

  2. $p(s|\pi,A)$: By the chain rule of probability

  $$p(s|\pi,A) = p(s_1|\pi,A)\prod_{t=2}^{T} p(s_t|s_1,\dots,s_{t-1},\pi,A)$$

  This is *always* true. By the *first-order Markov property* (i.e., the model definition) we can further say that $p(s_t|s_1,\dots,s_{t-1},\pi,A) = p(s_t|s_{t-1},\pi,A)$ and also simplify the conditioning by removing π or A as required.

  3. Notice that we've used indicators again to pick out the correct entries in π, A and B. This is basically always useful for discrete variables/data.

- EM steps: Recall the three main EM steps (the first two are the "E" and the third is the "M" step)

  1. Set $q(s) = p(s|x,H)$

  2. Calculate $\mathcal{L} = \mathbb{E}_q[\ln p(x,s|H)]$

  3. Maximize $\mathcal{L}$ over $H = \{\pi, A, B\}$

- The first step already poses a problem. Recall from the GMM that we had no difficulty learning $q(c)$ where c was the vector of cluster indicators ($c_n = j$ means observation x_n came from cluster j; refer to the previous lecture notes for more details). This is because conditioned on the GMM model variables θ, and the data x, $c_1,\dots,c_N$ were conditionally independent. Therefore, $p(c|x,\theta) = \prod_{n=1}^{N} p(c_n|x_n,\theta)$ and thus $q(c) = \prod_{n=1}^{N} q(c_n)$ where $q(c_n) = p(c_n|x_n,\theta)$.

- The conditional posterior $p(s|x,H) \neq \prod_{t=1}^{T} p(s_t|x_t,H)$, and so we can't easily solve for $q(s)$ in this way.

- There are K^T different sequences that s could be. In principal we would therefore have to calculate $p(s|x,H) \propto p(x|s,H)p(s|H)$ for each of these. Even though K^T is finite—and so we know that $p(s|x,H)$ is just a multinomial that can be calculated in principal—in practice K^T is way too large to enumerate all values.

- We'll see that we don't actually have to calculate $q(s)$ for each and every sequence s. However, we don't know that yet. Since we can find $q(s)$ in principal, we will address Step 1 above by simply declaring that we've calculated it.

- **Step 1 of EM**: "We have set $q(s) = p(s|x, H)$." Let's pretend this is the case to try and make some progress (and see if we actually don't need this to be the case).

- **Step 2 of EM**: Next take the expectation, $\mathbb{E}_q[\ln p(x, s|H)]$.

$$\begin{aligned}
\mathcal{L} \;=\; & \sum_{t=1}^{T}\sum_{k=1}^{K}\sum_{v=1}^{V}\mathbb{E}_q[\mathbb{1}(s_t = k)]\mathbb{1}(x_t = v)\ln B_{kv} && \leftarrow \mathbb{E}_q[\ln p(x|s, B)] \\
& + \sum_{k=1}^{K}\mathbb{E}_q[\mathbb{1}(s_1 = k)]\ln \pi_k && \leftarrow \mathbb{E}_q[\ln p(s_1|\pi)] \\
& + \sum_{t=2}^{T}\sum_{i=1}^{K}\sum_{j=1}^{K}\mathbb{E}_q[\mathbb{1}(s_{t-1} = i, s_t = j)]\ln A_{ij} && \leftarrow \mathbb{E}_q[\ln p(s_2, \dots, s_T|A)] \qquad (3)
\end{aligned}$$

- There are two key expectations here.

  1. $\mathbb{E}_q[\mathbb{1}(s_t = k)]$: The expectation of an indicator of an event is the probability of that event, $\mathbb{E}_q[\mathbb{1}(s_t = k)] = q(s_t = k)$. To see this,

$$\begin{aligned}
\sum_s q(s)\mathbb{1}(s_t = k) &= \sum_{s_1=1}^{K}\cdots\sum_{s_{t-1}=1}^{K}\sum_{s_{t+1}=1}^{K}\cdots\sum_{s_T=1}^{K} q(s_1, \dots, s_{t-1}, s_t = k, s_{t+1}, \dots, s_T) \\
&= p(s_t = k|x, H) \\
&\equiv q(s_t = k) \qquad (4)
\end{aligned}$$

     What is going on here? This expectation is just the *marginal* distribution of $q(s)$ where we integrate (i.e., sum) over every value of $s_{t'}$ except s_t which we set equal to k. In other words, instead of needing to know the entire posterior distribution of a sequence s, we are here only asking for the posterior probability that $s_t = k$ given x and H without regard to what any other value of the sequence is equal to.

  2. $\mathbb{E}_q[\mathbb{1}(s_{t-1} = i, s_t = j)]$: By using exactly the same reasoning as above,

$$\mathbb{E}_q[\mathbb{1}(s_{t-1} = i, s_t = j)] = p(s_{t-1} = i, s_t = j|x, H) \equiv q(s_{t-1} = i, s_t = j) \qquad (5)$$

     Again, we only care about the posterior probability of a specific transition at a specific time, not caring about anything else going on in the sequence.

- We still don't know what these probabilities are, or how to calculate them. However, we have shown that in order to complete the E-step, we never need to know the posterior distribution of the entire sequence. We only need to know marginal posterior distributions on isolated portions of that sequence.

- That is, we don't need to calculate $p(s|x, H)$, only $p(s_t = k|x, H)$ and $p(s_{t-1} = i, s_t = j|x, H)$ for all t, k, i and j. This is still non-trivial, but we'll see that there is an algorithm that let's us get these values quickly called the *forward-backward algorithm.*

- **Step 3 of EM**: Let's pretend we have $p(s_t = k|x, H)$ and $p(s_{t-1} = i, s_t = j|x, H)$ already calculated and quickly take care of the M-step. Using Lagrange multipliers to ensure that the updates for π, A and B are probability distributions, we can maximize

$$\mathcal{L} = \sum_{t,k,v} q(s_t = k)\mathbb{1}(x_t = v) \ln B_{kv} + \sum_k q(s_1 = k) \ln \pi_k + \sum_{t>1,i,j} q(s_{t-1} = i, s_t = j) \ln A_{ij} \quad (6)$$

as follows:

$$\pi_k = q(s_1 = k) \quad (7)$$

$$A_{ij} = \frac{\sum_{t=2}^T q(s_{t-1} = i, s_t = j)}{\sum_{k=1}^K \sum_{t=2}^T q(s_{t-1} = i, s_t = k)} \quad (8)$$

$$B_{kv} = \frac{\sum_{t=1}^T q(s_t = k)\mathbb{1}(x_t = v)}{\sum_{w=1}^V \sum_{t=1}^T q(s_t = k)\mathbb{1}(x_t = w)} \quad (9)$$

Calculating $q(s_t = k)$ and $q(s_{t-1} = i, s_t = j)$

- So we now just need to find these two marginal distributions. We will do this in a two-step procedure, first defining them in terms of quantities that again we wish we had, and then presenting an algorithm to find those quantities.

- Calculate $q(s_t = k)$: For this problem we want to set

$$\begin{aligned} q(s_t = k) &= p(s_t = k|x, H) \quad (10) \\ &\propto \underbrace{p(x_{t+1}, \dots, x_T|s_t = k, H)}_{\equiv \beta_t(k)} \underbrace{p(s_t = k|x_1, \dots, x_t, H)}_{\equiv \alpha_t(k)} \end{aligned}$$

- We make the definitions:

  - $\beta_t(k) = p(x_{t+1}, \dots, x_T|s_t = k, H)$, the probability of seeing everything to come after step t given that we are in state k at step t.

  - $\alpha_t(k) = p(s_t = k|x_1, \dots, x_t, H)$, the posterior probability of being in state k at step t given all the data observed up to, and including, that time point.

- We have simply used Bayes rule and define the two terms in the likelihood and prior as β and α. We could have used Bayes rule in other ways, for example using x_{t+2} and later in β and including x_{t+1} in α. The reason we don't is because we wouldn't be able to get things to work out that way. Ultimately, the goal is to use the rules of probability to keep re-writing things until we reach a form where we can actually start plugging in numbers and find solutions.

- Again we've pushed the solution off to a later point, but notice that, if we could find a way to calculate $\alpha_t(k)$ and $\beta_t(k)$, we could then set

$$q(s_t = k) = \frac{\beta_t(k)\alpha_t(k)}{\sum_{j=1}^{K} \beta_t(j)\alpha_t(j)} \tag{11}$$

- Calculate $q(s_{t-1} = i, s_t = j)$: For this one we again use Bayes rule, followed by additional factorizations,

$$\begin{aligned}
q(s_{t-1} = i, s_t = j) &= p(s_{t-1} = i, s_t = j|x, H) \qquad (12) \\
&\propto p(x_t, \dots, x_T|s_{t-1} = i, s_t = j, H)p(s_{t-1} = i, s_t = j|x_1, \dots, x_{t-1}, H) \\
&\propto p(x_t, \dots, x_T|s_t = j, H)p(s_t = j|s_{t-1} = i, H)p(s_{t-1} = i|x_1, \dots, x_{t-1}, H) \\
&\propto \underbrace{p(x_{t+1}, \dots, x_T|s_t = j, H)}_{\equiv \beta_t(j)}\underbrace{p(x_t|s_t = j, H)}_{B_{j,x_t}} \\
&\quad \times \underbrace{p(s_t = j|s_{t-1} = i, H)}_{A_{ij}}\underbrace{p(s_{t-1} = i|x_1, \dots, x_{t-1}, H)}_{\equiv \alpha_{t-1}(i)}
\end{aligned}$$

- Again, we have used Bayes rule in such a way that we can make progress towards solving for this posterior distribution. Notice that two of the terms are directly from the HMM variables. We can directly plug these values in for the most recent iteration of the algorithm. We don't know the other two probabilities, however, notice that they are exactly what we need to know for $q(s_t = k)$. Therefore, we can use the same definitions and whatever algorithm we develop to solve $q(s_t = k)$, we can use the same result to solve $q(s_{t-1} = i, s_t = j)$.

- Imagining that we have α and β, we can then set

$$q(s_{t-1} = i, s_t = j) = \frac{\beta_t(j)B_{j,x_t}A_{ij}\alpha_{t-1}(i)}{\sum_{r=1}^{K}\sum_{s=1}^{K} \beta_t(r)B_{r,x_t}A_{sr}\alpha_{t-1}(s)} \tag{13}$$

- Think of $q(s_{t-1} = i, s_t = j)$ as the (i, j)-th element in a $K \times K$ matrix that gives the probability of this transition. Since there can only be one transition, the sum of probabilities in this matrix has to equal one. The denominator above makes this be the case.

The forward-backward algorithm

- The algorithm that gives us α_t is called the "forward algorithm" while that which gives β_t is the "backward algorithm." They are recursive algorithms, meaning that to learn α_t we need α_{t-1}, and to learn β_t we need β_{t+1} (hence the directions in the name). Since we can find α_1 and β_T, we can solve for all α_t and β_t.

- Forward algorithm: Here we want to learn the K-dimensional vector α_t at time t, where $\alpha_t(k) = p(s_t = k|x_1, \dots, x_t, H)$. To do this, we first write this as the marginal of a joint probability distribution and then use Bayes rule on the joint distribution,

$$
\begin{aligned}
\underbrace{p(s_t = k|x_1, \dots, x_t, H)}_{\alpha_t(k)} &= \sum_{j=1}^{K} p(s_t = k, s_{t-1} = j|x_1, \dots, x_t) && (14)\\
&\propto \sum_{j=1}^{K} p(x_t|s_t = k, s_{t-1} = j, H) p(s_t = k, s_{t-1} = j|x_1, \dots, x_{t-1}, H)\\
&\propto \sum_{j=1}^{K} \underbrace{p(x_t|s_t = k, H)}_{B_{k,x_t}} \underbrace{p(s_t = k|s_{t-1} = j, H)}_{A_{jk}} \underbrace{p(s_{t-1} = j|x_1, \dots, x_{t-1}, H)}_{\equiv\, \alpha_{t-1}(j)}
\end{aligned}
$$

- Therefore, we can set

$$
\begin{aligned}
\hat{\alpha}_t(k) &= B_{k,x_t} \sum_{j=1}^{K} A_{jk} \alpha_{t-1}(j) && (15)\\
\alpha_t(k) &= \frac{\hat{\alpha}_t(k)}{\sum_{j=1}^{K} \hat{\alpha}_t(j)} && (16)
\end{aligned}
$$

- We also notice that we can solve

$$q(s_1 = k) = p(s_1 = k|x_1, H) = \alpha_1(k)$$

exactly using Bayes rule,

$$\alpha_1(k) = \frac{\pi_k B_{k,x_1}}{\sum_{j=1}^{K} \pi_j B_{j,x_1}} \qquad (17)$$

This means we know the starting point for α_1 and can solve all subsequent α_t.

- Backward algorithm: We now want to learn the K-dimensional vector β_t at time t, where $\beta_t(k) = p(x_{t+1}, \dots, x_T|s_t = k, H)$. We again write this as the marginal of a joint probability distribution. However, this time we won't need Bayes rule.

$$
\begin{aligned}
\underbrace{p(x_{t+1}, \dots, x_T|s_t = k, H)}_{\beta_t(k)} &= \sum_{j=1}^{K} p(x_{t+1}, \dots, x_T, s_{t+1} = j|s_t = k, H) && (18)\\
&= \sum_{j=1}^{K} p(x_{t+1}, \dots, x_T|s_{t+1} = j, H) p(s_{t+1} = j|s_t = k, H)\\
&= \sum_{j=1}^{K} \underbrace{p(x_{t+2}, \dots, x_T|s_{t+1} = j, H)}_{\equiv\, \beta_{t+1}(j)} \underbrace{p(x_{t+1}|s_{t+1} = j, H)}_{B_{j,x_{t+1}}} \underbrace{p(s_{t+1} = j|s_t = k, H)}_{A_{kj}}
\end{aligned}
$$

- Notice that this time there is no proportionality. We can just set

$$\beta_t(k) = \sum_{j=1}^{K} \beta_{t+1}(j) B_{j,x_{t+1}} A_{kj} \qquad (19)$$

- For the first value, we can then set $\beta_T(j) = 1$ for all j then solve for the previous values. Notice that these numbers can become very small as t becomes less and less leading to computer precision issues. After updating each β_t, you can normalize this vector. Notice that this re-scaling will not change any other updated values.

- Log marginal likelihood: To evaluate convergence, we need to calculate

$$\ln p(x_1, \dots, x_T|H) = \ln \sum_s p(x, s| \dots, H) \tag{20}$$

- Recall that since there are K^T different sequences s to sum over, we simply aren't going to calculate this marginal directly.

- However, using the chain rule of probability we can make progress. That is,

$$p(x_1, \dots, x_T|H) = p(x_1|H) \prod_{t=2}^{T} p(x_t|x_1, \dots, x_{t-1}, H) \tag{21}$$

- Notice that $p(x_t|x_1, \dots, x_{t-1}, H)$ is actually something we've calculated in the forward algorithm,

$$p(x_t|x_1, \dots, x_{t-1}, H) = \sum_{k=1}^{K} \hat{\alpha}_t(k) \tag{22}$$

- We see this by simply plugging in what we defined to be $\hat{\alpha}_t(k)$,

$$\begin{aligned} \sum_{k=1}^{K} \hat{\alpha}_t(k) &= \sum_{k=1}^{K} \sum_{j=1}^{K} p(x_t|s_t = k, H)p(s_t = k|s_{t-1} = j, H)p(s_{t-1} = j|x_1, \dots, x_{t-1}, H) \\ &= p(x_t|x_1, \dots, x_{t-1}, H) \end{aligned} \tag{23}$$

- Therefore,

$$\ln p(x_1, \dots, x_T|H) = \sum_{t=1}^{T} \ln \sum_{k=1}^{K} \hat{\alpha}_t(k) \tag{24}$$

- We can literally compute this value "in passing" during the forward algorithm.

Multiple sequences

- Finally, we show (but don't derive) how to modify the algorithm when we have multiple sequences $x^{(1)}, \dots, x^{(N)}$ of possibly varying length $T_1, \dots, T_N$. Hopefully this is a straightforward exercise to derive on your own by now.

- Forward-backward: First, given the HMM variables $H = \{\pi, A, B\}$ forward-backward is run *independently* on each sequence to obtain values $\alpha_t^{(n)}(k)$ and $\beta_t^{(n)}(k)$ for the nth sequence.

- $q(s_t^{(n)} = k)$ and $q(s_{t-1}^{(n)} = i, s_t^{(n)} = j)$: For sequence number n, these probabilities are computed using $\alpha_t^{(n)}(k)$ and $\beta_t^{(n)}(k)$ and π, A, B exactly as before.

- Updating π, A and B: Simply sum over the sequences and normalize

$$\pi_k = \frac{1}{N}\sum_{n=1}^{N} q(s_1^{(n)} = k) \tag{25}$$

$$A_{ij} = \frac{\sum_{n=1}^{N}\sum_{t=2}^{T_n} q(s_{t-1}^{(n)} = i, s_t^{(n)} = j)}{\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{t=2}^{T_n} q(s_{t-1}^{(n)} = i, s_t^{(n)} = k)} \tag{26}$$

$$B_{kv} = \frac{\sum_{n=1}^{N}\sum_{t=1}^{T_n} q(s_t^{(n)} = k)\mathbb{1}(x_t^{(n)} = v)}{\sum_{n=1}^{N}\sum_{w=1}^{V}\sum_{t=1}^{T_n} q(s_t^{(n)} = k)\mathbb{1}(x_t^{(n)} = w)} \tag{27}$$

MAP-EM inference

- In the last part, we will briefly see how small modifications can be made to derive MAP-EM and variational inference algorithms. In both cases we need to define priors on the HMM parameters, π, A and B. Typically conjugacy motivates these to be

$$A_{k,:} \sim \text{Dirichlet}(\alpha), \quad B_{k,:} \sim \text{Dirichlet}(\gamma), \quad \pi \sim \text{Dirichlet}(\kappa). \tag{28}$$

- For MAP-EM, we simply add $\ln p(H) = \ln p(\pi)\prod_k p(A_{k,:})p(B_{k,:})$ to the EM objective. In this case the update to $p(s)$ is identical to the previous one and the only difference is in the updates to π, A and B above, where terms are added in the numerator and denominator.

- I don't write those updates down, but point out that there are many -1's involved, one in the numerator and K of them in the denominator (so a $-K$ in the denominator) resulting from the Dirichlet prior. This indicates that for MAP-EM to be guaranteed to be well-defined, we need to set all Dirichlet parameters ≥ 1. The MAP solution gives a significantly different meaning to the Dirichlet parameters than what we *a priori* think (where sparsity requires them to be < 1).

- MAP for multinomial parameters with Dirichlet priors in general is a problem. Fortunately VI is easy in this case and fixes this interpretability problem. I'll focus on VI below.

Variational inference

- We want to approximate the posterior of the discrete HMM with Dirichlet priors using variational inference. We therefore first need to define a factorized q distribution. We will used

$$q(s, \pi, A, B) = q(s)q(\pi)\prod_{k=1}^{K} q(A_{k,:})q(B_{k,:}). \tag{29}$$

- Notice that we aren't factorizing over the values in the sequence s. However, the same exact expectations of identities show up here as well, so we can follow the same reasoning to arrive at the same algorithm as for ML-EM (very slightly modified as described below).

- Let's pretend we know $q(s)$. Then the following standard (by now) steps can be followed:

$$\begin{aligned} q(A_{k,:}) &\propto \exp\Big\{\sum_{t=2}^{T}\sum_{j=1}^{K} q(s_{t-1}=k, s_t=j)\ln A_{k,j} + (\alpha-1)\ln A_{k,j}\Big\} \\ &= \text{Dirichlet}\Big(\Big[\alpha + \sum_{t=2}^{T} q(s_{t-1}=k, s_t=j)\Big]_{j=1}^{K}\Big) \end{aligned} \tag{30}$$

- Where the $[\,]$ notation indicates a vector ranging over values of j. Similarly,

$$q(B_{k,:}) = \text{Dirichlet}\Big(\Big[\gamma + \sum_{t=1}^{T} q(s_t=k)\mathbb{1}(x_t=v)\Big]_{v=1}^{V}\Big) \tag{31}$$

$$q(\pi) = \text{Dirichlet}\Big(\Big[\kappa + q(s_1=k)\Big]_{k=1}^{K}\Big) \tag{32}$$

- If there are multiple sequences, the s variables have a sequence index and all q distributions above include an extra summation over the sequence index.

- What about $q(s)$? We won't go into all details, but simply point out that the following changes can be made to the forward-backward algorithm.

$$A_{ij} \rightarrow e^{\mathbb{E}_q[\ln A_{ij}]}, \quad B_{kv} \rightarrow e^{\mathbb{E}_q[\ln B_{kv}]}, \quad \pi_i \rightarrow e^{\mathbb{E}_q[\ln \pi_i]} \tag{33}$$

- This follows the familiar pattern. We also recognize that all expectations are using Dirichlet q distributions, so are of the form $\psi(\cdot) - \psi(\sum \cdot)$. It doesn't impact the algorithm that these modifications don't sum to one and so aren't probability distributions anymore—they're simply plugged in.

- Finally, to calculate the variational objective function $\mathcal{L}$, we note that

$$\begin{aligned} \mathcal{L} &= \int q(\pi)\ln\frac{p(\pi)}{q(\pi)}d\pi + \sum_{i=1}^{K}\int q(A_{i,:})\ln\frac{p(A_{i,:})}{q(A_{i,:})}dA_{i,:} + \sum_{k=1}^{K}\int q(B_{k,:})\ln\frac{p(B_{k,:})}{q(B_{k,:})}dB_{k,:} \\ &\quad + \sum_{t=1}^{T}\ln\sum_{k=1}^{K}\hat{\alpha}_t(k) \end{aligned} \tag{34}$$

- Notice that the term involving $\hat{\alpha}_t(k)$ is just like before, only we've replaced A with $e^{\mathbb{E}_q[\ln A]}$, etc., when calculating it. The other three terms, are the negative KL-divergences between the approximate posterior q and prior p of each variable.

- Finally, to calculate $\mathcal{L}$ when there are multiple sequences, the term involving $\hat{\alpha}_t$ has an additional sequence index and a second sum is done over this index. The first line is unchanged.