Graphical-Duration Hidden Markov Model

László Keresztes, *ELTE TTK*
Supervisor: Balázs Csanád Csáji, *SZTAKI*, *ELTE*

I. Hidden Markov Models

A Hidden Markov Model (HMM) could be viewed as a noisy observation of a Markov chain. This model emerged in the 1960s, and now it has important applications in signal processing, control theory, speech recognition and sequential bioinformatics [4], [5]. In the HMM framework, there is a hidden Markov process that influences the observations, but we cannot observe it directly. Usually, the inference for this hidden process is the task to solve, where the hidden process is our real process of interest, such as a sequence of words in speech recognition or specific DNA regions in the DNA sequence. An HMM has a transition model and an observation model. The transition model controls the hidden process, at each time step, we stochastically move to the next hidden state. The observation model tells us how the observations are generated from a hidden state. Each hidden state has a data generating distribution, these distributions came from a parametric family, such as Gaussians or Categorical distributions. The parametric family should be selected in advance, based on a priori knowledge or empirical data distribution.

The transition model is a Markov chain, which could be viewed as a directed graph. The structure of the graph could be chosen according to domain expert knowledge. Building these expert thoughts correctly into the model makes it more reasonable, more robust, and less prone to error.

The next step would be to chose a parametric family for duration distributions by the experts, and building these information into the model.

A. Structure

In my thesis, I deal with discrete-time, finite-state Hidden Markov Models. The theorems and proofs are designed for the categorical observation model, but the ideas apply for any other observation model.

Definition 1. *(Discrete-time Hidden Markov Model)*
Let Z_t and X_t discrete-time stochastic processes with $t \geq 1$. The pair (Z_t, X_t) is a Hidden Markov Model if:

1) *Z_t is a Markov process (that cannot be observed directly)*
2) $P(X_t \in B | Z_s = z_s \ s \geq 1, X_s = x_s \ s \neq t) = P(X_t \in B | Z_t = z_t)$

We call an HMM finite-state if there is only finitely many values that Z_t could take. We can assume, that $Z_t \in \{1, \ldots, M\}$, where $1, \ldots, M$ are the possible hidden states. We also assume, that the HMM is time-homogeneous ($p(z_t = j | z_{t-1} = i)$ and $p(x_t | z_t = i)$ are independent of t).

So an HMM is a hidden process, a discrete $z_t \in \{1, \ldots, N\}$ Markov chain in discrete time ($t \in \{1, \ldots, T\}$), and an observation model $p(x_t | z_t)$. The joint distribution has the form

$$p(z_{1:T}, x_{1:T}) = p(z_1) \prod_{t=2}^{T} p(z_t | z_{t-1}) \prod_{t=1}^{T} p(x_t | z_t)$$

The initial distribution $\pi_i = p(z_1 = i)$ is a probability distribution on $\{1, \ldots, N\}$.

The transition model $A_{ij} \doteq p(z_t = j | z_{t-1} = i)$ is independent of the time t (time-homogeneous). A is an $N \times N$ matrix, also called the transition matrix.

The observation model could represent discrete or continuous distributions. In the discrete case the observation model is a matrix of B, where $B_{il} = p(x_t = l | z_t = i)$ for the $l = 1, \ldots, L$ categories and for the $i = 1, \ldots, N$ hidden states. In the continuous case there is usually a parametric family, such as Gaussians: $p(x_t | z_t = i) = \mathcal{N}(x_t | \mu_i, \Sigma_i)$, where the conditional distribution has the parameters μ_i and Σ_i.

In the next chapters, we will consider HMMs with categorical observation model ($x_1, \ldots, x_T \in \{1, \ldots, L\}$). The HMM has parameters $\theta = (\pi, A, B)$.

The most basic inference tasks are filtering, smoothing, and MAP estimation.

In filtering we want to compute (online) the $\alpha_t(i) = p(z_t = i | x_{1:t})$ belief state which could be done by the forward algorithm. The forward algorithm is a forward DP algorithm.

In smoothing we want to compute (offline) the $\gamma_t(i) = p(z_t = i | x_{1:T})$ given all the data and this could be done by the forward algorithm and the backward algorithm. In the backward algorithm we compute $\beta_t(j) = p(x_{t+1:T} | z_t = j)$. The backward algorithm is a backward DP, and then $\gamma_t(j) \propto \alpha_t(j)\beta_t(j)$ could be get.

In learning, besides filtering and smoothing, computing the two-slice marginals $\xi_{t,t+1}(i, j) = p(z_t = i, z_{t+1} = j | x_{1:T})$ is also essential. This could be done as $\xi_{t,t+1}(i, j) \propto \alpha_t(i) A_{ij} \beta_{t+1}(j) p(x_{t+1} | z_{t+1} = j)$ from the already computed α, β values.

The MAP (maximum a posteriori) estimation is the computation of

$$\arg\max_{z_{1:T}} p(x_{1:T} | z_{1:T})$$

This could be done with an offline, forward DP also known as Viterbi decoding.

II. EM learning

Learning in HMM means we want to learn the starting probabilities $p(z_1)$, the transition probabilities $p(z_t | z_{t-1})$ and the parameters of the observation model.

Because of the usually unobservable hidden process, we cannot maximize directly the likelihood function, therefore an iterative approach called Expectation-Maximization is applied.

A. EM learning in general

The idea of EM is the following. We usually want to maximize the log likelihood of the observed data:

$$l(\theta) = \log p(x_{1:T} | \theta) = \log \Big[\sum_{z_{1:T}} p(x_{1:T}, z_{1:T} | \theta) \Big]$$

This is hard to optimize, therefore instead we maximize the complete data log likelihood:

$$l_c(\theta) = \log p(x_{1:T}, z_{1:T} | \theta)$$

This cannot be computed, since z_t are unknown. Define the expected complete data log likelihood as the following:

$$Q(\theta; \theta^{n-1}) = E[l_c(\theta) | x_{1:T}, \theta^{n-1}] = E_{z_{1:t} | x_{1:t}, \theta^{n-1}} [l_c(\theta)] = E_{z_{1:T} \sim p(z_{1:T} | x_{1:T}, \theta^{n-1})} [l_c(\theta)]$$

Here, the z_t are replaced with their expected value conditioned on the data and the previous parameter set.

The idea of the EM is that since we do not know the actual values of z_t, starting from an initial guess of parameters, we can iteratively estimate z_t with probabilities from the parameters (and data), then estimate the parameters using the z_t estimates.

The condition is usually on the amount of gain in the Q function or the number of iterations.

The EM algorithm in general finds a local optimum (with certain assumptions) by increasing the observed data log-likelihood at every EM step. [1], [3]

```
Algorithm 1: Expectation-Maximization (EM) algorithm
Input  : Observation sequence $x_{1:T}$,
         initial parameters $\theta^0$
Output: Parameters $\theta^N$
Until condition:
    • E step: Compute $Q(\theta; \theta^{n-1})$ or the expected sufficient
      statistics (for parameter update)
    • M step:
          $\theta^n = \arg\max_{\theta} Q(\theta; \theta^{n-1})$
```

Statement 1. *EM increases the observed data log likelihood*
For the (θ^n) parameter series from the EM algorithm:

$$l(\theta^{n+1}) \geq l(\theta^n)$$

Proof. Denote $X = x_{1:T}$, $Z = z_{1:T}$. Denote the distribution $q^n(Z) = p(Z | X, \theta^n)$. Let D denote the information divergence, and H the entropy function.

$$\begin{aligned} l(\theta) &= \log p(X | \theta) = \log p(X, Z | \theta) - \log p(Z | X, \theta) \\ &= \sum_Z q^n(Z) \log p(X, Z | \theta) - \sum_Z q^n(Z) \log p(Z | X, \theta) \\ &= Q(\theta; \theta^n) - \sum_Z q^n(Z) \log \Big[\frac{p(Z | X, \theta)}{q^n(Z)} q^n(Z) \Big] \\ &= Q(\theta; \theta^n) + D(q^n(Z) || p(Z | X, \theta)) + H(q^n(Z)) \end{aligned}$$

This is true for every θ. Now setting $\theta = \theta^n$:

$$\begin{aligned} l(\theta^n) &= Q(\theta^n; \theta^n) + D(q^n(Z) || p(Z | X, \theta^n)) + H(q^n(Z)) \\ &= Q(\theta^n; \theta^n) + H(q^n(Z)) \end{aligned}$$

By differentiating the two equations, we have:

$$\begin{aligned} l(\theta) - l(\theta^n) &= Q(\theta; \theta^n) - Q(\theta^n; \theta^n) + D(q^n(Z) || p(Z | X, \theta)) \\ &\geq Q(\theta; \theta^n) - Q(\theta^n; \theta^n) \end{aligned}$$

Selecting

$$\theta^{n+1} = \arg\max_{\theta} Q(\theta, \theta^n)$$

shows that $l(\theta^{n+1}) \geq l(\theta^n)$. □

One of the best practices is to use multiple randomized initializations of the EM algorithm and select the best parameters.

In some cases (e.g. with HMM) both the E-step and M-step have an analytical solution. This could be also true with different parameter constraints: e.g. with parameter tying, re-parameterization.

B. EM learning for HMMs - Baum-Welch algorithm

Applying the EM algorithm for learning HMM parameters, the complete data log likelihood is simply the log of the joint:

$$l_c(\theta) = \log p(z_1 | \theta) + \sum_{t=2}^{T} \log p(z_t | z_{t-1}, \theta) + \sum_{t=1}^{T} \log p(x_t | z_t, \theta)$$

The auxiliary $Q(\theta; \theta^n)$ function has the following form:

$$\begin{aligned}
Q(\theta; \theta^n) &= E_{\underline{z} \sim p(\underline{z}|\underline{x}, \theta^n)}[l_c(\theta)] \\
&= E_{z_1 \sim p(z_1|\underline{x}, \theta^n)}[\log p(z_1|\theta)] + \\
&+ \sum_{t=2}^{T} E_{(z_{t-1}, z_t) \sim p((z_{t-1}, z_t)|\underline{x}, \theta^n)}[\log p(z_t|z_{t-1}, \theta)] + \\
&+ \sum_{t=1}^{T} E_{z_t \sim p(z_t|\underline{x}, \theta^n)}[\log p(x_t|z_t, \theta)] \\
&= \sum_{i=1}^{M} \log \pi_i \cdot p(z_1 = i|\underline{x}, \theta^n) + \\
&+ \sum_{t=2}^{T} \sum_{i=1}^{M} \sum_{j=1}^{M} \log A_{ij} \cdot p(z_{t-1} = i, z_t = j|\underline{x}, \theta^n) + \\
&+ \sum_{t=1}^{T} \sum_{i=1}^{M} \sum_{l=1}^{L} \log B_{il} \mathbb{I}(x_t = l) \cdot p(z_t = i|\underline{x}, \theta^n) \\
&= \sum_{i=1}^{M} \log \pi_i \gamma_1^n(i) + \sum_{t=2}^{T} \sum_{i=1}^{M} \sum_{j=1}^{M} \log A_{ij} \xi_{t-1,t}^n(i, j) + \\
&+ \sum_{t=1}^{T} \sum_{i=1}^{M} \sum_{l=1}^{L} \log B_{il} \mathbb{I}(x_t = l) \gamma_t^n(i)
\end{aligned}$$

The E step involves the computation of the expected sufficient statistics:

- $\gamma_t^n(i) = p(z_t = i|x_{1:T}, \theta^n)$
- $\xi_{t-1,t}^n(i, j) = p(z_{t-1} = i, z_t = j|x_{1:T}, \theta^n)$

Conditioning on θ^n means computing the γ and ξ values on the HMM with parameters θ^n. As we already see, the γ and ξ values could be computed with dynamic programming algorithms.

The M step involves constrained optimization: we want to optimize in π, A, B, but we must ensure that:

- π is a probability distribution on $\{1, \ldots, M\}$
- $\forall i \; A_{i,:}$ is a probability distribution on $\{1, \ldots, M\}$
- $\forall i \; B_{i,:}$ is a probability distribution on $\{1, \ldots, L\}$

We could optimize separately in π, $A_{i:}$ for $i = 1, \ldots, M$, and $B_{i,:}$ for $i = 1, \ldots, M$.

Statement 2. *(M step optimization as divergence minimization)*
Let $a_i \geq 0$ for $i = 1, \ldots, M$. The probability distribution p on $\{1, \ldots, M\}$, that maximizes

$$\sum_{i=1}^{M} \log p_i \cdot a_i$$

is $p_i = a_i/a$, if $a = \sum_{i=1}^{M} a_i > 0$.

Proof. If $a_i = 0 \; \forall i$, then any p maximizes the term. Note, that the following proof returns correctly that $a_i = 0 \implies p_i = 0$.

Define the probability distribution $\hat{a}$ with $\hat{a}_i = \frac{a_i}{a}$.

$$\begin{aligned}
\arg\max_p \sum_{i=1}^{M} \log p_i \cdot a_i &= \arg\max_p \sum_{i=1}^{M} \log p_i \cdot \hat{a}_i \\
&= \arg\max_p \sum_{i=1}^{M} \hat{a}_i \log p_i - \sum_{i=1}^{M} \hat{a}_i \log \hat{a}_i \\
&= \arg\max_p -D(\hat{a}||p)
\end{aligned}$$

We have $-D(\hat{a}||p) \leq 0$ and equality if and only if $p = \hat{a}$. □

For the θ^{n+1} updated parameters:

$$\begin{aligned}
\pi^{n+1} &= \arg\max_{\pi} \sum_{i=1}^{M} \log \pi_i \gamma_1^n(k) = \gamma_1^n \\
A_{i,:}^{n+1} &= \arg\max_{A_{i,:}} \sum_{t=2}^{T} \sum_{j=1}^{M} \log A_{ij} \xi_{t-1,t}^n(i, j) \\
&= \arg\max_{A_{i,:}} \sum_{j=1}^{M} \log A_{ij} \Big(\sum_{t=2}^{T} \xi_{t-1,t}^n(i, j) \Big) \\
&\propto \Big(\sum_{t=2}^{T} \xi_{t-1,t}^n(i, j) \Big)_{j=1,\ldots,M} \\
B_{i,:}^{n+1} &= \arg\max_{B_{i,:}} \sum_{t=1}^{T} \sum_{l=1}^{L} \log B_{il} \mathbb{I}(x_t = l) \gamma_t^n(i) \\
&= \arg\max_{B_{i,:}} \sum_{l=1}^{L} \log B_{il} \Big(\sum_{t=1}^{T} \mathbb{I}(x_t = l) \gamma_t^n(i) \Big) \\
&\propto \Big(\sum_{t=1}^{T} \mathbb{I}(x_t = l) \gamma_t^n(i) \Big)_{l=1,\ldots,L}
\end{aligned}$$

The results are quite intuitive:

- $\pi_i^{n+1} \propto \gamma_1(i)^n$
- $A_{ij}^{n+1} \propto \sum_{t=2}^{T} \xi_{t-1,t}^n(i, j)$
- $B_{il}^{n+1} \propto \sum_{t=1}^{T} \gamma_t^n(i) \mathbb{I}(x_t = l)$

These are all expected counts on the corresponding events. The EM learning in the HMM framework is called the Baum-Welch algorithm.

Statement 3. *(Zero persistency in EM)*
If we initialize the EM algorithm with such θ^0, that has $A_{ij}^0 = 0$, then:

$$\forall n : A_{ij}^n = 0$$

.

Proof. It is enough to show that $A_{ij}^1 = 0$.

From the computation of ξ, we know that if $A_{ij}^0 = 0$, then $\xi_{t-1,t}^0(i, j) = 0$ for $t = 2, \ldots, T$. But $A_{ij}^1 \propto \sum_{t=2}^{T} \xi_{t-1,t}^0(i, j) = 0$, which shows that $A_{ij}^1 = 0$. □

C. Complexity of the Baum-Welch algorithm

One iteration of the Baum-Welch algorithm involves an E-step and an M-step computation for the HMM. Now assume, that E is the edge number of the θ^0 initialized HMM ($E \geq M - 1$).

As we already know, the E-step is the computation of γ and ξ values and that takes $\mathcal{O}(TM^2)$ time or $\mathcal{O}(TE)$ time in a sparse graph.

On the time complexity of the M-step: for the update of π we need $\mathcal{O}(M)$ time. For the update of A, we need to update at M^2 or E places, and each takes $\mathcal{O}(T)$ time.

For the B matrix, we have $M \times L$ parameters, for each it takes $\mathcal{O}(T)$ time to update. But for each $l \in \{1, \ldots, L\}$ we only need to sum over $T_l = \{t : x_t = l\}$: $B_{il} \propto \sum_{t \in T_l} \gamma_t(i)$. So for each $i \in \{1, \ldots, M\}$, the complexity is $\sum_{l=1}^{L} |T_l| = T$, because T_l is a partition of the $\{1, \ldots, T\}$ indicies.

So the full time complextiy of an M-step is $\mathcal{O}(M + TM^2 + TM) = \mathcal{O}(TM^2)$ or $\mathcal{O}(M + TE + TM) = \mathcal{O}(TE)$.

Therefore one iteration of the Baum-Welch takes $\mathcal{O}(TM^2)$ time or $\mathcal{O}(TE)$ time. As we already see, E does not increase with the Baum-Welch algorithm. So the initial number of edges E strongly affects the time complexity of the Baum-Welch.

III. Graph representation of distributions

The notation $p(v|u)$ for u, v (hidden) states is only the short form of the time independent $p(z_t = v|z_{t-1} = u)$.

One main setback of HMMs is that in general, each hidden state i has a duration $T_i \sim Geo(p_i)$. The geometric distribution corresponds to the most simple graph: vertices are $\{r, v_1, s\}$, edges are $\{(r, v_1), (v_1, v_1), (v_1, s)\}$ with $p(v_1|r) = 1$, $p(v_1|v_1) = p$ and also $p(s|v_1) = 1 - p$. The first arrival to the vertex s (starting from r at index 0) signs the transition to another state. One could extend the graph with $p(s|s) = 1$ to ensure a stochastic transition matrix and therefore a Markov chain (but this does not alter the computation). So given this graph, the probability that the first arrival to s is at step $d + 1$ is

$$P(\inf\{k : x_k = s\} = d + 1) = (1 - p)p^{d-1} = Geo(p)(d)$$

for the $(x)_k$ Markov chain starting from $x_0 = r$. The duration $d \geq 1$, which refers to the same logic as in graphical models, if we step into a state, we must spend 1 time-unit there (in discrete time).

The generalization of the previous idea (representing duration distributions with graphs) is possible.

A. Representation graphs

Formalizing the occurred concepts:

Definition 2. *(Duration distribution)*
Let $X : \Omega \to \mathbb{N}_+$ be random variable. Then $T = p_X$, the distribution of X is a duration distribution.

Examples for duration distributions: geometric distribution, categorical distribution on $\{1, \ldots, D\}$, negative binomial distribution. A mixture of duration distributions is also a duration distribution. The Poisson distribution is not a duration distribution, but if we truncate it to $[1, \infty)$ and normalize it (to integrate to 1), we get a duration distribution (call it Poisson duration distribution).

Definition 3. *(Parametric family of duration distributions)*
Let Θ be a parameter space. If for every $\theta \in \Theta$: $X(\theta) : \Omega \to \mathbb{N}_+$, then $\{T(\theta) : \theta \in \Theta\} = \{p_{X(\theta)} : \theta \in \Theta\}$ is a parametric family of duration distributions.

Examples for parametric family of duration distributions: geometric distributions with parameter p, categorical distributions on $\{1, \ldots, D\}$ with parameters $p_1, \ldots, p_D$, negative binomial distributions with parameters N, p, negative binomial distributions of fixed order N with parameter p, Poisson duration distributions with parameter λ.

One could think of learning the probabilities of self-transitions in the HMM framework as, given the family of geometric distributions, we should learn p. That is, similar to the observation model, a family is given. So, if the duration comes from a geometric family, it is fine. But what if we know that the duration comes from another family? Such as $Cat(\{1, \ldots, D\})$?

It will be shown that some duration distribution families could be represented as graphs, and in the next chapter, it would be introduced that one could "merge" these graphs to form a "two-layer" HMM with state durations from the desired family. There are many possible representations, therefore we should measure the "efficiency" of the representation.

Definition 4. *(Representation graph)*
A $G(\eta)$ Markov chain is a representation graph if we have r, s nodes that:

1) r is the starting node with probability 1
2) $G(\eta)$ stays in r only at index 0
3) s is the ending node with probability 1

For a representation graph the following properties hold:

1) $r, v_1, \ldots, v_n, s$ are the nodes
2) r is the starting node with probability 1
3) $G(\eta)$ stays in r only at index 0
4) s is the ending node with probability 1 ($P(\inf\{k : x_k = s\} < \infty) = 1$)
5) $p(r|r) = 0$, $p(s|r) = 0$, $p(s|s) = 1$
6) $\forall i : p(r|v_i) = 0$
7) $\exists i : p(s|v_i) > 0$
8) $E(G) = E_{fix}(G) \dot{\cup} E_{prob}(G)$, where the probabilites in E_{fix} are fixed 0s or $1s$, and the probabilities in E_{prob} are fully controlled by η

The indexing starts from 0 for a $G(\eta)$ sample and the number of steps taken in $G(\eta)$ (or the duration) for a sample is d, if the first arrival to s is at $d + 1$.

Denote the distribution of duration from $G(\eta)$ generated samples with $T[G(\eta)]$.

If we denote two representation graphs with $G(\eta_1)$ and $G(\eta_2)$ it means that they have the same structure, only the probabilities on the non-fixed edges could differ.

Formally, if $X_0, X_1, \ldots$ is the Markov chain $G(\eta)$ with $X_0 = r$, then:

$$\begin{aligned}
T[G(\eta)](d) &= P(\inf\{k : X_k = s\} = d + 1) \\
&= P(X_{d+1} = s, X_d \neq s) \\
&= P(X_{d+1} = s \text{ first time}) \\
&= P(X_{d+1} = s \text{ ft}) = P_{G(\eta)}(X_{d+1} = s \text{ ft})
\end{aligned}$$

The first example of the geometric distribution is a $G(p)$ representation graph with $E_{fix} = \{(r, v_1)\}$ and $E_{prob} = \{(v_1, v_1), (v_1, s)\}$. As we already observed, $T[G(p)] = Geo(p)$.

Definition 5. *(Properties of a representation graph)*
Let $G(\eta)$ be a representation graph. Then:

- $e_{in} \doteq |\{i : p(v_i|r) \not\equiv 0\}|$ *the number of incoming edges*
- $e_{out} \doteq |\{i : p(s|v_i) \not\equiv 0\}|$ *the number of outgoing edges*
- $e \doteq |\{i, j : p(v_j|v_i) \not\equiv 0\}|$ *the number of inner edges*
- $n \doteq |V(G)| - 2$ *the number of nodes*
- $V_{inn} \doteq \{v_1, \ldots, v_n\}$ *the set of inner nodes*

An edge (u, v) is $p(v|u) \not\equiv 0$ in this definition, if $(u, v) \in E_{fix}(G)$ with probability 1 or if $(u, v) \in E_{prob}(G)$.

The geometric distribution representation graph $G(p)$ has the following edge numbers: $e_{in} = 1$, $e_{out} = 1$, $e = 1$. The number of nodes is $n = 1$. □

Definition 6. *(Graph representation of duration distribution)*

Let T be a duration distribution. Let $G(\eta)$ be a representation graph. $G(\eta)$ represents T if $T = T[G(\eta)]$.

Definition 7. *(Graph representation of duration distribution families)*

Let $T(\theta)$ be a parametric family of duration distributions. Let $\{G(\eta) : \eta \in H\}$ be a family of representation graphs based on the same structure.

G represents $T(\theta)$ (the family) if

$$\forall\theta\ \exists\eta\ T(\theta) = T[G(\eta)]$$
$$\forall\eta\ \exists\theta\ T[G(\eta)] = T(\theta)$$

For example, the family of geometric distributions with parameter p could be represented with the same graph structure as at the beginning of the chapter, only with different $\eta = p$ values.

The main question is how other distribution families could be represented with graphs.

Example: consider the representation graph $G(p)$ with nodes r, v_1, v_2, v_3, s and with the following non-zero probabilities:

- $p(v_1|r) = 1$
- $p(v_1|v_1) = p$
- $p(v_2|v_1) = 1 - p$
- $p(v_2|v_2) = p$
- $p(v_3|v_2) = 1 - p$
- $p(v_3|v_3) = p$
- $p(s|v_3) = 1 - p$

It is not hard to see, that G represents the family of negative binomial distributions of fixed order 3. [3]

Statement 4. *(Walk-based description)*

Let $X_0, X_1, \ldots$ be the Markov chain of the $G(\eta)$ representation graph. Let $W_{d+1} = \{x_0, x_1, \ldots, x_{d+1} : x_0 = r, x_{d+1} = s, x_i \neq s\ \forall i \leq d\}$ denote the set of $r \to s$ walks with length $d + 1$ (and without s as an inner point). Then:

$$P(X_{d+1} = s \text{ ft}) = \sum_{w \in W_{d+1}} \prod_{e \in w} p(e)$$

Proof. The form of the Markov chain indicates that $\{X_d \neq s\} = \{X_i \neq s\ \forall i \leq d\}$.

$$\begin{aligned}
P(X_{d+1} = s \text{ ft}) &= \sum_{\substack{x_0,\ldots,x_{d+1} \\ x_0 = r, x_{d+1} = s \\ x_d \neq s}} P(X_0 = x_0, \ldots, X_{d+1} = x_{d+1}) \\
&= \sum_{\substack{x_0,\ldots,x_{d+1} \\ x_0 = r, x_{d+1} = s \\ x_d \neq s}} \prod_{j=1}^{d+1} p(x_j|x_{j-1}) \\
&= \sum_{w \in W_{d+1}} \prod_{e \in w} p(e)
\end{aligned}$$

□

B. Representation of distribution families

The following duration distribution families have a graph representation: geometric family with parameter p, negative binomial distributions of fixed order N with parameter p, categorical distributions on $\{1, \ldots, D\}$ with parameters $p_1, \ldots, p_D$.

Statement 5. *(Representation of geometric family)*

The $Geo(p)$ geometric family could be represented by a $G(p)$ graph with nodes r, v_1, s and with the following non-zero probabilities:

- $p(v_1|r) = 1$
- $p(v_1|v_1) = 1 - p$
- $p(s|v_1) = p$

Proof. We know that $Geo(p)(d) = (1 - p)^{d-1}p$ for $d \geq 1$. Using the definition of $T[G(p)]$:

$$\begin{aligned}
T[G(p)](d) &= P(\inf\{k : x_k = s\} = d + 1) \\
&= P(X_0 = r, X_1 = v_1, \ldots, X_d = v_1, X_{d+1} = s) \\
&= P(X_0 = r) \cdot P(X_1 = v_1|X_0 = r) \cdot \prod_{i=2}^{d} P(X_i = v_1|X_{i-1} = v_1) \cdot \\
&\quad \cdot P(X_{d+1} = s|X_d = v_1) \\
&= 1 \cdot p(v_1|r) \cdot \prod_{i=2}^{d} p(v_1|v_1) \cdot p(s|v_1) \\
&= 1 \cdot 1 \cdot (1 - p)^{d-1} \cdot p = (1 - p)^{d-1}p
\end{aligned}$$

There is a clear bijection between $Geo(p)$ instances and $G(p)$ instances; using the same p. □

Statement 6. *(Representation of negative binomial family of fixed order N)*

The $NB_N(p)$ negative binomial family could be represented by a $G(p)$ graph with nodes r, $v_1, \ldots, v_N$, s and with the following non-zero probabilities:

- $p(v_1|r) = 1$
- $p(v_i|v_i) = 1 - p$ *for* $i = 1, \ldots, N$
- $p(v_i|v_{i-1}) = p$ *for* $i = 2, \ldots, N$
- $p(s|v_N) = p$

Proof. We know that $NB_N(p)(d) = \binom{d-1}{N-1}(1 - p)^{d-N}p^N$ for $d \geq N$.

We prove by induction.

For $N = 1$, this is the geometric distribution and the previous statement.

For $N > 1$, assume we know the statement for $N - 1$. By separating on the first arrival to v_N, and using the induction step:

$$\begin{aligned}
T[G(p)](d) &= P(X_{d+1} = s \text{ ft}) \\
&= \sum_{i=N}^{d} P(X_{d+1} = s \text{ ft}, X_i = v_N \text{ ft}) \\
&= \sum_{i=N}^{d} P(X_i = v_N \text{ ft})P(X_{d+1} = s \text{ ft} \mid X_i = v_N \text{ ft}) \\
&= \sum_{i=N}^{d} NB_{N-1}(p)(i - 1)Geo(p)(d - i + 1) \\
&= \sum_{i=N}^{d} \binom{i-2}{N-2}(1 - p)^{i-N}p^{N-1}(1 - p)^{d-i}p \\
&= (1 - p)^{d-N}p^N \sum_{i=N}^{d} \binom{i-2}{N-2} \\
&= (1 - p)^{d-N}p^N \Big[\binom{N-2}{N-2} + \sum_{i=N+1}^{d} \binom{i-1}{N-1} - \binom{i-2}{N-1}\Big] \\
&= (1 - p)^{d-N}p^N \binom{d-1}{N-1}
\end{aligned}$$

There is a clear bijection between $NB_N(p)$ instances and $G(p)$ instances; using the same p.

□

Statement 7. *(Representation of categorical distributions on $\{1, \ldots, D\}$)*

The $Cat(\{1, \ldots, D\})$ categorical family with parameters $p_1, \ldots, p_D$ could be represented by a $G(p_1, \ldots, p_D)$ graph with nodes r, $v_1, \ldots, v_D$, s and with the following non-zero probabilities:

- $p(v_1|r) = 1$
- $p(v_d|v_1) = p_{D+2-d}$ *for* $d = 2, \ldots, D$
- $p(v_d|v_{d-1}) = 1$ *for* $d = 3, \ldots, D$
- $p(s|v_D) = 1$
- $p(s|v_1) = p_1$

Proof. We know that the $Cat(\{1, \ldots, D\})$ distribution has $p_1, \ldots, p_D$ parameters with $\sum_{d=1}^{D} p_d = 1$ and $p_d \geq 0$. $Cat(\{1, \ldots, D\})(d) = p_d$ simply.

We will use the walk-based description: $P(X_{d+1} = s \text{ ft}) = \sum_{w \in W_{d+1}} \prod_{e \in w} p(e)$.

For $d = 1$: $W_2 = \{(r, v_1, s)\}$, therefore $P(X_2 = s \text{ ft}) = p(v_1|r)p(s|v_1) = p_1$.

For $2 \leq d \leq D$: $W_{d+1} = \{(r, v_1, v_{D+2-d}, \ldots, v_D, s)\}$, therefore

$$\begin{aligned}
P(X_{d+1} = s \text{ ft}) &= p(v_1|r) \cdot p(v_{D+2-d}|v_1) \cdot \prod_{j=D+2-d}^{D-1} p(v_{j+1}|v_j) \cdot p(s|v_D) \\
&= p_{D+2-(D+2-d)} = p_d
\end{aligned}$$

For $d > D$: $W_{d+1} = \emptyset$, therefore $P(X_{d+1} = s \text{ ft}) = 0$. □

In the next chapter, we will see two more graph representations for $Cat\{1, \ldots, D\}$.

It is not hard to see that the mixture distributions could be represented if all the individuals could be represented.

Statement 8. *(Representation of mixture distributions)*

Let the $\{T_i(\theta_i) : \theta_i \in \Theta_i\}$ family represented by a $G_i(\eta_i)$ graph for $i = 1, 2$. Then the family $\{\rho T_1(\theta_1) + (1 - \rho)T_2(\theta_2) : \rho \in [0, 1], \theta_1 \in \Theta_1, \theta_2 \in \Theta_2\}$ could be represented by a graph $G(\rho, \eta_1, \eta_2)$ with nodes r, $V_{inn}(G_1)$, $V_{inn}(G_2)$, s and with the following non-zero probabilities:

- $p(v_i^1|r) = \rho \cdot p_{G_1(\theta_1)}(v_i^1|r)$ *for* $v_i^1 \in V_{inn}(G_1)$
- $p(v_i^2|r) = (1 - \rho) \cdot p_{G_2(\theta_2)}(v_i^2|r)$ *for* $v_i^2 \in V_{inn}(G_2)$
- $p(v_j^1|v_i^1)$, $p(s|v_i^1)$ *as in* $G_1(\theta_1)$
- $p(v_j^2|v_i^2)$, $p(s|v_i^2)$ *as in* $G_2(\theta_2)$

Proof. It is enough to prove that

$$T[G(\rho, \eta_1, \eta_2)] = \rho T[G_1(\eta_1)] + (1 - \rho)T[G_2(\eta_2)]$$

Denote $P_{G(\rho,\eta_1,\eta_2)}$ with P for brevity.

$$\begin{aligned}
T[G(\rho, \eta_1, \eta_2)] &= P(X_{d+1} = s \text{ ft}) \\
&= P(X_{d+1} = s \text{ ft}|X_1 \in V_{inn}(G_1))P(X_1 \in V_{inn}(G_1)) + \\
&\quad + P(X_{d+1} = s \text{ ft}|X_1 \in V_{inn}(G_2))P(X_1 \in V_{inn}(G_2)) \\
&= \rho P_{G_1(\eta_1)}(X_{d+1} = s \text{ ft}) + (1 - \rho)P_{G_2(\eta_1)}(X_{d+1} = s \text{ ft}) \\
&= \rho T[G_1(\eta_1)] + (1 - \rho)T[G_2(\eta_2)]
\end{aligned}$$

□

Although, not every distribution family and not every distribution could be represented.

Statement 9. *(Non-representation of light-tailed distributions)*

Let T a duration distribution with infinite support and with the following property:

$$\limsup_{d\to\infty} \frac{T(d)}{\alpha^d} = 0 \quad \forall\alpha > 0$$

Then there is no finite graph that could represent the distribution T.

Proof. Assume that $G(\eta)$ represents T.

If $G(\eta)$ has no positive circle, then it could only represent a finite-support distribution. (Because in this case, the nodes form a DAG, so a topological order exists, and the maximum length of an rs walk is $n(G(\eta)) + 1$.)

Let d_0 large enough ($d_0 > n(G(\eta)) + 1$), and consider the walk-based description:

$$T(d_0) = P(X_{d_0+1} = s \text{ ft}) = \sum_{w \in W_{d_0+1}} \prod_{e \in w} p(e)$$

Select a $w \in W_{d_0+1}$ positive walk; there must be at least one circle in this walk (otherwise it would not have length d_0). Select a circle from the walk, and name it C. Denote the walk before C with w_0 and the walk after C with w_1.

So $w = w_0 C w_1$, and let $c = |C|$ be the length of C (i.e. the number of edges). Use the notation $p_w = \prod_{e \in w} p(e)$ for any walk w, then we have:

$$\begin{aligned} T(d_0) &\geq \prod_{e \in w} p(e) \\ &= \prod_{e \in w_0} p(e) \prod_{e \in C} p(e) \prod_{e \in w_1} p(e) \\ &= p_{w_0} p_C p_{w_1} > 0 \end{aligned}$$

Define the following series:

$$d_j = d_0 + cj, \; j = 0, 1, \ldots$$

Then for $j \geq 0$; the walk $w^j = w_0 C^{j+1} w_1$ is a positive, d_j-length walk, so:

$$T(d_j) \geq p_{w_0} p_C^{j+1} p_{w_1} > 0$$

Let $\alpha < p_C^{1/c}$, then:

$$\begin{aligned} \limsup_{d \to \infty} \frac{T(d)}{\alpha^d} &\geq \limsup_{j \to \infty} \frac{T(d_j)}{\alpha^{d_j}} \\ &= \limsup_{j \to \infty} \frac{T(d_j)}{\alpha^{d_0 + cj}} \\ &\geq \lim_{j \to \infty} \frac{p_{w_0} p_C^{j+1} p_{w_1}}{\alpha^{d_0} \alpha^{cj}} \\ &= \frac{p_{w_0} p_C p_{w_1}}{\alpha^{d_0}} \lim_{j \to \infty} \Big(\frac{p_C}{\alpha^c}\Big)^j \\ &= \infty \end{aligned}$$

So the light-tailed property is violated, therefore no such $G(\eta)$ representation graph exists. □

Statement 10. *(Non-representation of Poisson duration distributions)*
Let T be one member of the Poisson duration distribution family.
Then there is no finite graph that could represent the distribution T.

Proof. We have $T(d) = C \frac{\lambda^d}{d!}$, so T has infinite-support and T is light-tailed, therefore the previous statement applies. □

IV. Graphical-Duration Hidden Markov Model

The GD-HMM is a simple HMM with parameter tyings and reparameterization. The model builds up from a simple HMM structure and replaces the "nodes" with the desired graph, that represents the duration family.

Consider the (π, A, θ_o) HMM model with M hidden states, where π is the initial distribution, A is the transition matrix and θ_o is the observation parameter matrix. For simplicity, we assume that $A_{ii} = 0$ for all i.

Let $\{T_i(\theta_i)\}$ a parametric family of duration distributions, represented with the $\{G_i(\eta_i)\}$ family.

For the graph G_i, use the following notations:

- $D_i = n(G_i)$ the number of (inner) nodes
- $r_i = r(G_i)$ starting node
- $s_i = s(G_i)$ ending node
- $\{i_1, \ldots, i_{D_i}\} = V_{inn}(G_i)$
- $e^i_{in} = e_{in}(G_i)$ the number of incoming edges
- $e^i_{out} = e_{out}(G_i)$ the number of outgoing edges
- $e^i = e(G_i)$ the number of inner edges

The $G_i(\eta_i)$ graph is still a Markov chain on nodes $r_i, i_1, \ldots, i_{D_i}, s_i$ with transition probabilities $p_{G_i(\eta_i)}(v|u)$ for u, v (hidden) states.

Definition 8. *(GD-HMM)*
Let (π, A, θ_o) be an HMM model with M hidden states, and T_i is a duration distribution, represented with $G_i(\eta_i)$ $\forall i = 1, \ldots, M$.
The GD-HMM is a $(\tilde{\pi}, \tilde{A}, \tilde{\theta}_o)$ HMM model.
For $i = 1, \ldots, M$:

- *hidden states: $i_d \in V_{inn}(G_i)$ for $d = 1, \ldots, D_i$*
- *transition probabilities*
  - *$\tilde{A}(i_k, i_l) \doteq p_{G_i(\eta_i)}(i_l | i_k)$ for $k, l = 1, \ldots, D_i$*
  - *$\tilde{A}(i_k, j_l) \doteq p_{G_i(\eta_i)}(s_i | i_k) A_{ij} p_{G_j(\eta_j)}(j_l | r_j)$ for $k = 1, \ldots, D_i$ for $l = 1, \ldots, D_j$ for $j \neq i$*
- *initial distribution $\tilde{\pi}(i_1) = \pi(i)$, $\tilde{\pi}(i_k) = 0$ for $k = 2, \ldots, D_i$*
- *observation model parameters $\tilde{\theta}_o(i_k) = \theta_o(i)$ for $k = 1, \ldots, D_i$*

The parameters of the GD-HMM are $(\pi, A, \theta_o, (\eta_1, \ldots, \eta_M))$.

If we want to build a GD-HMM from an HMM with $A_{ii} > 0$, then in the computation of $\tilde{A}(i_k, j_l)$, we should work with $\frac{A_{ij}}{1 - A_{ii}}$ instead of A_{ij}.

Statement 11. *(The GD-HMM is an HMM)*

$$\sum_v \tilde{A}(i_k, v) = 1$$

Proof.

$$\begin{aligned} \sum_v \tilde{A}(i_k, v) &= \sum_{l=1}^{D_i} \tilde{A}(i_k, i_l) + \sum_{\substack{j=1 \\ j \neq i}}^{M} \sum_{l=1}^{D_j} \tilde{A}(i_k, j_l) \\ &= \sum_{l=1}^{D_i} p_{G_i(\eta_i)}(i_l | i_k) + \sum_{\substack{j=1 \\ j \neq i}}^{M} \sum_{l=1}^{D_j} p_{G_i(\eta_i)}(s_i | i_k) \frac{A_{ij}}{1 - A_{ii}} p_{G_j(\eta_j)}(j_l | r_j) \\ &= 1 - p_{G_i(\eta_i)}(s_i | i_k) + p_{G_i(\eta_i)}(s_i | i_k) \sum_{\substack{j=1 \\ j \neq i}}^{M} \frac{A_{ij}}{1 - A_{ii}} \sum_{l=1}^{D_j} p_{G_j(\eta_j)}(j_l | r_j) \\ &= 1 \end{aligned}$$

□

The GD-HMM has two layers of representation: a lower-level representation with i_d, which forms a Markov chain, and a higher-level representation with $i \leftrightarrow \{i_1, \ldots, i_{D_i}\}$, which corresponds to the original hidden states.

The number of (non-zero) edges in a GD-HMM is:

$$E = \sum_{i=1}^{M} e^i + \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}}^{M} e^i_{out} e^j_{in} \mathbb{I}(A_{ij} > 0)$$

The number of (non-zero) edges in a dense GD-HMM (when the original HMM is complete) is:

$$E = \sum_{i=1}^{M} e^i + \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}}^{M} e^i_{out} e^j_{in}$$

The number of nodes is $V = \sum_{i=1}^{M} D_i$. The number of parameters in a GD-HMM could be upper-bounded by V (initial distribution) + E (real transitions) + VL (observation parameters).

If we assume that all $D_i = D$ are equal, and $e^i = \mathcal{O}(D)$, $e^i_{in} = \mathcal{O}(1)$ and $e^i_{out} = \mathcal{O}(1)$, then the number of nodes is MD and the number of edges is $\mathcal{O}(MD + M^2)$, which results in a sparse graph if $D \gg M$.

(HMM is a subclass of the GD-HMM)

Let (π, A, θ_o) be an HMM with M hidden states. Let $T_i = Geo(1 - p_i)$, and $\forall i = 1, \ldots, M$ consider the representation graph $G_i(p_i)$ with nodes r_i, i_1, s_i and the following non-zero probabilities:

- $p(i_1 | r_i) = 1$
- $p(i_1 | i_1) = p_i$
- $p(s_i | i_1) = 1 - p_i$

The resulting GD-HMM is a $(\pi, \tilde{A}, \theta_o)$ HMM model on the $\{1, \ldots, M\}$ nodes with:

$$\tilde{A}_{ij} = \begin{cases} (1 - p_i) A_{ij} / (1 - A_{ii}) & \text{if } j \neq i \\ p_i & \text{if } j = i \end{cases}$$

This gives back the original HMM if $p_i = A_{ii}$ $\forall i$.

V. Learning the parameters of GD-HMM

In the previous section, a new HMM variant was presented, but because of its special properties, we must go through the Baum-Welch algorithm to see what steps need to be modified.

Assume that the initialization is correct, i.e. we construct the θ^0 GD-HMM from a (π^0, A^0, B^0) HMM with $A^0_{ii} = 0$ and from the $G_i(\eta^0_i)$ graphs as in the definition. The initialized GD-HMM has $E = \sum_{i=1}^{M} e^i + \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}}^{M} e^i_{out} e^j_{in} \mathbb{I}(A^0_{ij} > 0)$ edges. We already see, that E does not increase during the EM.

As the model is still an HMM, the E-step (forwards-backwards algorithm) including every related computation could be done as before: $\alpha, \beta, \gamma, \xi$. The time complexity is $\mathcal{O}(TE)$ as we already see. Also, the Viterbi decoding could be done as before as well.

However, the M-step must be changed, because, from the definition of GD-HMM: no individual update on $A_{i,:}$ probabilities allowed. Here, the reformulation of EM (Baum-Welch) algorithm is presented:

The auxiliary function $Q(\theta; \theta^n)$ for a simple HMM on $\{1, \ldots, M\}$ nodes has the following form:

$$\begin{aligned} Q(\theta; \theta^n) = \sum_{i=1}^{M} \log \pi_i \gamma^n_1(i) + \sum_{t=2}^{T} \sum_{i=1}^{M} \sum_{j=1}^{M} \log A_{ij} \xi^n_{t-1,t}(i, j) + \\ + \sum_{t=1}^{T} \sum_{i=1}^{M} \sum_{l=1}^{L} \log B_{il} \mathbb{I}(x_t = l) \gamma^n_t(i) \end{aligned}$$

The GD-HMM has nodes $\{i_k : k \in \{1, \ldots, D_i\}, i \in \{1, \ldots, M\}\}$:

$$\begin{aligned} Q(\theta; \theta^n) = \sum_{i=1}^{M} \sum_{k=1}^{D_i} \log \tilde{\pi}_{i_k} \gamma^n_1(i_k) + \sum_{t=2}^{T} \sum_{i=1}^{M} \sum_{k=1}^{D_i} \sum_{j=1}^{M} \sum_{l=1}^{D_j} \log \tilde{A}_{i_k, j_l} \xi^n_{t-1,t}(i_k, j_l) + \\ + \sum_{t=1}^{T} \sum_{i=1}^{M} \sum_{k=1}^{D_i} \sum_{l=1}^{L} \log \tilde{B}_{i_k, l} \mathbb{I}(x_t = l) \gamma^n_t(i_k) \end{aligned}$$

We need to rewrite the auxiliary function to a function of $(\pi, A, B, (\eta_1, \ldots, \eta_M))$. Use the short notations $p_i = p_{G_i(\eta_i)}$, $\xi(i_k, j_l) = \sum_{t=2}^{T} \xi^n_{t-1,t}(i_k, j_l)$, $T_l = \{t : x_t = l\}$. We rewrite the function term by term:

Initial distribution:

$$\sum_{i=1}^{M} \sum_{k=1}^{D_i} \log \tilde{\pi}_{i_k} \gamma^n_1(i_k) = \sum_{i=1}^{M} \log \pi_i \gamma^n_1(i_1)$$

Transition probabilities:

$$\begin{aligned} &\sum_{t=2}^{T} \sum_{i=1}^{M} \sum_{k=1}^{D_i} \sum_{j=1}^{M} \sum_{l=1}^{D_j} \log \tilde{A}_{i_k, j_l} \xi^n_{t-1,t}(i_k, j_l) = \\ &\sum_{i=1}^{M} \sum_{k=1}^{D_i} \sum_{l=1}^{D_i} \log p_i(i_l | i_k) \xi(i_k, i_l) + \\ &\sum_{i=1}^{M} \sum_{k=1}^{D_i} \sum_{\substack{j=1 \\ j \neq i}}^{M} \sum_{l=1}^{D_j} \log(p_i(s_i | i_k) A_{ij} p_j(j_l | r_j)) \xi(i_k, j_l) = \\ &\sum_{i=1}^{M} \sum_{k=1}^{D_i} \Bigg[\sum_{l=1}^{D_i} \log p_i(i_l | i_k) \xi(i_k, i_l) + \log p_i(s_i | i_k) \Big(\sum_{\substack{j=1 \\ j \neq i}}^{M} \sum_{l=1}^{D_j} \xi(i_k, j_l) \Big) \Bigg] + \\ &\sum_{i=1}^{M} \Bigg[\sum_{\substack{j=1 \\ j \neq i}}^{M} \log A_{ij} \Big(\sum_{k=1}^{D_i} \sum_{l=1}^{D_j} \xi(i_k, j_l) \Big) \Bigg] + \\ &\sum_{j=1}^{M} \Bigg[\sum_{l=1}^{D_j} \log p_j(j_l | r_j) \Big(\sum_{\substack{i=1 \\ i \neq j}}^{M} \sum_{k=1}^{D_i} \xi(i_k, j_l) \Big) \Bigg] \end{aligned}$$

Emission probabilities:

$$\sum_{t=1}^{T}\sum_{i=1}^{M}\sum_{k=1}^{D_i}\sum_{l=1}^{L}\log\tilde{B}_{i_k,l}\mathbb{I}(x_t=l)\gamma_t^n(i_k)=$$
$$\sum_{i=1}^{M}\left[\sum_{l=1}^{L}\log B_{il}\Big(\sum_{t=1}^{T}\sum_{k=1}^{D_i}\mathbb{I}(x_t=l)\gamma_t^n(i_k)\Big)\right]=$$
$$\sum_{i=1}^{M}\left[\sum_{l=1}^{L}\log B_{il}\Big(\sum_{t\in T_l}\sum_{k=1}^{D_i}\gamma_t^n(i_k)\Big)\right]$$

We see that an analytical update is possible in the M-step, because the Q function could be written as a sum of $\sum_{i\in I} a_i \log p_i$ terms, where $(p_i : i \in I)$ is a probability distribution and $a_i \geq 0\ \forall i \in I$.

The time-complexity of the M-step is:

- $\mathcal{O}(M+\sum_{i=1}^{M} D_i)$ for the initial distribution
- $\mathcal{O}(TE)$ for the transition probabilities $(A_{ij}, (\eta_1, \ldots, \eta_M))$:
    1) $\mathcal{O}(TE)$ for computing $\xi(i_k, j_l) = \sum_{t=2}^{T}\xi_{t-1,t}^n(i_k, j_l)$ values for $\{(i_k, j_l) : \tilde{A}_{i_k,j_l}^0 > 0\}$, the others are zeroes
    2) $\mathcal{O}(E)$ for computing $\sum_{\substack{j=1\\j\neq i}}^{M}\sum_{l=1}^{D_j}\xi(i_k, j_l)$ coefficients for all (i_k, s_i) exit edges: $\sum_{i=1}^{M} e_{out}^i \sum_{j=1} e_{in}^j \mathbb{I}(A_{ij}^0 > 0) \leq E$, because $\xi(i_k, j_l) > 0$ implies that (i_k, s_i) exit edge, $A_{ij}^0 > 0$ and (r_j, j_l) entry edge.
    3) $\mathcal{O}(E)$ for computing $\sum_{k=1}^{D_i}\sum_{l=1}^{D_j}\xi(i_k, j_l)$ coefficients for all $\{(i, j) : A_{ij} > 0\}$): similarly as previous
    4) $\mathcal{O}(E)$ for computing $\sum_{\substack{i=1\\i\neq j}}^{M}\sum_{k=1}^{D_i}\xi(i_k, j_l)$ coefficients for all (r_j, j_l) entry edges: similarly as previous
    5) $\mathcal{O}(E)$ for updating $p_i(i_l|i_k)$ and $p_i(s_i|i_k)$ parameters for all i_k: for each i, we have $e^i + e_{out}^i$ non-zero edges in G_i, $\sum_{i=1}^{M} e^i + e_{out}^i \leq E$
    6) $\mathcal{O}(E)$ for updating A_{ij} parameters for all i, j: $\sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M}\mathbb{I}(A_{ij} > 0) \leq E$
    7) $\mathcal{O}(E)$ for updating $p_j(j_l|r_j)$ parameters for all j_l: $\sum_{j=1}^{M} e_{in}^j \leq E$
    8) $\mathcal{O}(E)$ for assigning every non-zero (i_k, j_l) edge their new $\tilde{A}_{i_k,j_l} = p_i(s_i|i_k)A_{ij}p_j(j_l|r_j)$ probability and every (i_k, i_l) edge their new $\tilde{A}_{i_k,i_l} = p_i(i_l|i_k)$ probability
- $\mathcal{O}(T(M+\sum_{i=1}^{M} D_i))$ for the emission probabilities (B_{il}):
    1) $\mathcal{O}(T\sum_{i=1}^{M} D_i)$ for summing up γ values: $\gamma_t(i) \doteq \sum_{k=1}^{D_i}\gamma_t^n(i_k)$
    2) $\mathcal{O}(TM)$ for updating B_{il} parameters: $\sum_{i=1}^{M}\sum_{l=1}^{L}\log B_{il}\sum_{t\in T_l}\gamma_t(i)$ as in simple HMM (for all i: B_{il} needs $|T_l|$ additions)
    3) $\mathcal{O}(T\sum_{i=1}^{M} D_i)$ for assigning the corresponding emission probabilities: $\tilde{B}_{i_k,l} = B_{il}$

In summary, we have, that the M-step could be done in $\mathcal{O}(TE)$ time, such as in the simple Baum-Welch algorithm, and therefore one EM iteration for GD-HMM takes $\mathcal{O}(TE)$ time.

VI. Efficiency of representation in GD-HMM

As we have already seen, the number of (non-zero) edges is the key measure of the time complexity of the forwards-backwards algorithm (and EM algorithm) in any HMM.

We advance the usefulness of the number of edges and define efficiency of representation.

Definition 9. *(Representation efficiency of GD-HMM)*

Let $\theta = (\pi, A, \theta_O)$ is an HMM and let T_i be duration distributions represented with G_i graphs. The full efficiency of representation is the number of edges in the resulting GD-HMM:

$$E(\{G_i\}, \{T_i\}, \theta) = \sum_{i=1}^{M} e^i + \sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M} e_{out}^i e_{in}^j \mathbb{I}(A_{ij} > 0)$$

We would like to measure how efficient is the representation of T_i with G_i, so we should create a simpler definition of efficiency, that does not depend on the θ HMM. We could examine only the GD-HMMs from HMMs with complete graphs ($\forall i \neq j : A_{ij} > 0$).

Definition 10. *(Representation efficiency function)*

Let θ be an HMM with complete graph. Let T_i be duration distributions represented with G_i graphs. The efficiency-function of representation is $E : \mathbb{N}_+ \to \mathbb{N}_+$ defined by the following:

$$E(\{G_i\}, \{T_i\})(M) = \sum_{i=1}^{M} e^i + \sum_{i=1}^{M}\sum_{\substack{j=1\\j\neq i}}^{M} e_{out}^i e_{in}^j$$

Now we can measure the goodness of representations together. Next, we want to measure the efficiency of individual representations. The motivation is that each T_i may come from the same family. To succeed next we assume that every T_i is represented with $G(\eta_i)$, so the inner structure of the graph is the same.

Definition 11. *(Representation efficiency function of graphs)*

Let $\{T(\theta) : \theta \in \Theta\}$ is a parametric family of duration distributions. Let G is the representation graph of $\{T(\theta)\}$. The efficiency function of representation is the following:

$$E(G, \{T(\theta)\})(M) = Me(G) + M(M-1)e_{out}(G)e_{in}(G)$$

which is simply the narrowing of the previous definition to the case of G represents all T_i.

Remember, that the geometric distribution representation graph $G(p)$ has the following edge numbers: $e_{in} = 1$, $e_{out} = 1, e = 1$. Therefore the efficiency-function is $E(G(p), Geo(p))(M) = M+M(M-1) = M^2$ which is the number of edges in a complete HMM.

From the previous definition, it is clear that we want more efficient representations for duration distribution families; i.e. representation with fewer edges.

For example consider the family of categorical distributions on $\{1, \ldots, D\}$ with parameters $p_1, \ldots, p_D$. Here is the construction of three different graphs G_1, G_2, G_3 each of them represents the family, but with different efficiency.

Let G_1 has $D+2$ nodes and has the following non-zero probability transitions:

- $p(v_d|r) = p_{D+1-d}$ for $d = 1, \ldots, D$
- $p(v_d|v_{d-1}) = 1$ for $d = 2, \ldots, D$
- $p(s|v_D) = 1$

The efficiency is $M(D-1) + M(M-1)D$. This representation comes from Yu & Kobayashi [6].

Let G_2 has $D+2$ nodes and has the following non-zero probability transitions:

- $p(v_1|r) = 1$
- $p(v_d|v_1) = p_{D+2-d}$ for $d = 2, \ldots, D$
- $p(v_d|v_{d-1}) = 1$ for $d = 3, \ldots, D$
- $p(s|v_D) = 1$
- $p(s|v_1) = p_1$

The efficiency is $M(2D-3) + M(M-1)2$. This is more efficient than G_1 as long as $M \geq 2$ and $D \geq 2$. This representation was presented in the statement of categorical representation.

Let G_3 has $2+1+2+\ldots+D = D(D-1)/2+2$ nodes (endowed with double index) and has the following non-zero probability transitions:

- $p(v_{d,1}|r) = p_d$ for $d = 1, \ldots, D$
- $p(v_{d,k}|v_{d,k-1}) = 1$ for $k = 2, \ldots d$ for $d = 1, \ldots, D$
- $p(s|v_{d,d}) = 1$ for $d = 1, \ldots, D$

The efficiency is $M(D-1)(D-2)/2 + M(M-1)D^2$. This is the worst among the three.

The following statements tell us, that the second representation is optimal.

Statement 12. *(Optimal representation of categorical distributions)*

Let $\{T(\theta) : \theta \in \Theta\}$ is the family of categorical distributions on $\{1, \ldots, D\}$, with $\theta = (p_1, \ldots, p_D)$. Let G represents this family. Then

$$E(G, \{T(\theta)\})(M) \geq M(D-1) + M(M-1)$$

Proof. Reminder for the walk-based description:

$$P(X_{d+1} = s \text{ ft}) = \sum_{w\in W_{d+1}}\prod_{e\in w} p(e)$$

.

Let $p_D > 0$ and let $G(\eta)$ represent $T(p_1, \ldots, p_D)$. Then:

$$0 < p_D = P_{G(\eta)}(X_{D+1} = s \text{ ft}) = \sum_{w\in W_{D+1}}\prod_{e\in w} p_\eta(e)$$

Note: $p(e)$ only depends on η, because G is fixed. So we have at least one $D+1$-length $w = (r, x_1, \ldots, x_D, s)$ $r \to s$ walk with positive probability.

The $x_1, \ldots, x_D$ nodes are all inner nodes.

I claim they are all different. Assume, that $\exists i < j : x_i = x_j$. In this case, $C = x_i \cdots x_j$ is a positive circle. Denote the walk before C with w_0, and the walk after C with w_1. Let $c = |C| \geq 1$ the length of the circle. Then $\forall j \geq 1$: $w^j = w_0 C^{j+1} w_1$ is a positive walk with length $D+1+jc$. So $T[G(\eta)](D+jc) > 0\ \forall j$, but $T(p_1, \ldots, p_D)(D+jc) = 0$. Because $G(\eta)$ represents T, all nodes have to be different.

We have D different inner nodes: $x_1, \ldots, x_D$. Using the positive walk $w = (r, x_1, \ldots, x_D, s)$: $e_{in} \geq 1$, $e_{out} \geq 1$ and $e \geq D-1$. We have:

$$E(G, \{T(\theta)\})(M) = Me(G) + M(M-1)e_{out}(G)e_{in}(G) \geq M(D-1) + M(M-1)$$

□

Thus, the second representation has efficiency $\mathcal{O}(MD+M^2)$ and the optimal efficiency is also $\mathcal{O}(MD+M^2)$. The time complexity of forwards-backwards algorithm (and one iteration of EM) is $\mathcal{O}(TE)$ in a sparse graph. Now $E = MD+M^2 \ll (MD)^2$. This leads to an $\mathcal{O}(T(MD+M^2))$ complexity using the efficient representation for categorical family. No better time complexity could be achieved with a different representation and this complexity is the same as in [2], [6].

References

[1] Olivier Cappé, Eric Moulines, and Tobias Ryden. *Inference in Hidden Markov Models (Springer Series in Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2005.

[2] Carl Mitchell, Mary Harper, and Leah Jamieson. On the complexity of explicit duration hmms. 03 1999.

[3] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[4] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. pages 257–286, 1989.

[5] Shun-Zheng Yu. Hidden semi-markov models. *Artificial Intelligence*, 174(2):215–243, 2010. Special Review Issue.

[6] Shun-Zheng Yu and H. Kobayashi. An efficient forward-backward algorithm for an explicit-duration hidden markov model. *IEEE Signal Processing Letters*, 10(1):11–14, 2003.