Residential Time HMMs

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

I. Motivation

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, and sequential bioinformatics. 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 different DNA regions in the DNA sequence. Any HMM has a transition and an observation model. The observation model tells us how the observations are generated from a hidden state. It usually comes from a parametric family of probability distributions, such as Gaussians or Categorical distributions (but easily exchangeable for any other parametric family in any state). However, the transition model also enables us to have huge flexibility to build our a priori knowledge into the model. Building these expert thoughts into the model makes it more reasonable, more robust, and less prone to error. One possible information is the residential time in each state. This reformulation ends up in a fixed HMM model, but we should usually rewrite the learning algorithm, which is Expectation-Maximization (EM) in our case.

II. Hidden Markov Models

An HMM is a hidden process, a discrete $z_t \in \{1, \dots, N\}$ Markov chain in discrete time, 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 start probabilities $\pi_i = p(z_1 = i)$ is a probability distribution on $\{1, \dots, N\}$.

The transition model $A_{ij} \doteq p(z_t = j|z_{t-1} = i)$ is independent of the time t. 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 $B_{kl} = p(x_t = l|z_t = k)$ for $l = 1...L$ the categories and for $i = 1...N$ the hidden states. In the continuous case there is usually parameterized family of distributions, such as Gaussians: $p(x_t|z_t = k) = \mathcal{N}(x_t|\mu_k, \Sigma_k)$, where the conditional distribution has the parameters μ_k and Σ_k.

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 and 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 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.

III. EM learning in HMM

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.

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

$$l(\theta) = \sum_{t=1}^{T} \log p(x_t|\theta) = \sum_{t=1}^{T} \log \big[\sum_i p(x_t, z_t = i|\theta) \big]$$

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

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

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

$$Q(\theta, \theta^{n-1}) = E\big[l_c(\theta)|x_{1:T}, \theta^{n-1}\big]$$

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.

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:
          $\theta^n = \arg\max_{\theta} Q(\theta, \theta^{n-1})$
```

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

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

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

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

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

The conditioning on θ^{n-1} is simply that computing the γ and ξ values on the HMM with parameters θ^{n-1}.

The M step involves constrained optimization, we want to optimize in π, A and observation model parameters, but we must ensure that $\sum_i \pi_i = 1$ and $\pi_i \geq 0$, also that A is a stochastic matrix, and a similar constraint could apply for the model parameters.

In general case, fortunately, the optimization could be done separately in π, $A_{i:}$ for $i = 1, \ldots, N$ and observation model parameters for hidden state $i = 1, \ldots, N$.

The results are quite intuitive. Here the Categorical distribution is presented.

- $\hat{\pi}_i \propto \gamma_1(i)$
- $\hat{A}_{kj} \propto \sum_{t=2}^{T} \xi_{t-1,t}(k, j)$
- $\hat{B}_{kl} \propto \sum_{t=1}^{T} \gamma_t(k)\mathbb{I}(x_t = l)$

These are all expected counts on the corresponding events.

The EM algorithm in general finds a local optimum (with certain assumptions) by increasing the likelihood at every EM step. [1], [2]

The EM learning in the HMM framework is called the Baum-Welch algorithm.

IV. 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 reason is behind the graph structure of the Markov chain of z_t hidden states. The geometric distribution corresponds to the most simple graph/flow: 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 would be always the question, starting from r at index 0, but one could extend the graph with $p(s|s) = 1$ to ensure a stochastic transition matrix and therefore a Markov chain (but it does not matter on 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 durations with graphs) is possible. The possible terms for representation: graphs, flows, Markov chains are used here interchangeably.

Formalizing the occurred concepts:

Definition 1. *(Duration distribution)*
Let $X : \Omega \to \mathbb{N}_+$ be random variable. Then $T = p(X)$, the distribution of X is a duration distribution.

The terms probability mass function and distribution would be used interchangeably as long as the intention is clear. 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 2. *(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 distribution 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 models, 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 more than one possible representations, therefore we should measure the "efficiency" of the representation.

Definition 3. *(Representation graph)*

A $G(\eta)$ Markov chain is a representation graph if the following hold:

1) *$r, v_1, \dots, v_n, s$ are the nodes*
2) *r is the starting node with probability 1*
3) *s is the ending node with probability 1*
4) *$p(r|r) = 0$, $p(s|r) = 0$, $p(s|s) = 1$*
5) *$\forall i : p(r|v_i) = 0$*
6) *$\exists i : p(s|v_i) > 0$*
7) *$E(G) = E_{fix}(G) \dot\cup E_{prob}(G)$, 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.

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 $x_0, x_1, \dots$ is a sample generated from the Markov chain $G(\eta)$ with $x_0 = r$, then:

$$T[G(\eta)](d) = P(\inf\{k : x_k = s\} = d + 1)$$

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

Definition 4. *(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*
- *$V_{inn} \doteq \{v_1, \dots, v_n\}$ 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 edges number: $e_{in} = 1$, $e_{out} = 1$, $e = 1$. The number of nodes is $n(G(p)) = 1$.

Definition 5. *(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 6. *(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 and possibly different probability values.

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

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

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. [2]

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 $\{1, \dots, D\}$ with parameters $p_1, \dots, p_D$.

Statement 1. *(Representation of geometric family)*

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

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

Statement 2. *(Representation of negative binomial family of fixed order N)*

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

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

Statement 3. *(Representation of categorical distributions on $\{1, \dots, D\}$)*

The $Cat(\{1, \dots, D\})$ family could be represented by a $G(p_1, \dots, p_D)$ graph. In the next chapter we will see 3 different graph representations for $Cat\{1, \dots, D\}$.

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

Statement 4. *(Representation of mixture distributions)*

Let the $\{T_i(\theta_i) : \theta_i \in \Theta_i\}$ family represented by a $G_i(\theta_i)$ 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

$G(\rho,\theta_1,\theta_2)$ 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 $G_2(\theta_2)$*

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

Statement 5. *(Non-representation of light-tailed distributions)*

Let T a duration distribution 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.

Statement 6. *(Non-representation of Poisson duration distributions)*

Let T be one member of the Poisson duration distribution family. Then $T(d) = C\frac{\lambda^d}{d!}$, therefore the previous statement applies.

At the same time, approximate representation is possible to any degree for any distribution.

Statement 7. *(Approximate representation of distributions)*

Let T be any duration distribution. Then:

$$\exists D\, \exists S \in Cat(\{1,\dots,D\}) : |S - T| < \epsilon$$

Now S could be represented by a graph.

V. Residential time HMM

One could construct HMM-like models, that are aware of time, different options could be found in [3]. The variants are usually called Hidden Semi-Markov Model, Variable Duration Hidden Markov Model, or Explicit Duration Hidden Markov Model.

Each solution in the review of Yu introduces new graphical models with "counter states", and does not try to capture duration times inside the HMM framework.

The most simple solution from the review of Yu is the residential time HMM (RT-HMM) which assumes that a state transition is either $(i,1) \to (j,\tau)$ for $j \neq i$ or $(i,\tau) \to (i,\tau-1)$ where τ is the residential time of state i. [3], [4]

They provided the forward-backward algorithm for the model, a modification of the HMM's forward-backward algorithm. The algorithm takes $\mathcal{O}((M^2+MD)T)$ steps, where T is the length of the observation sequence, M is the number of hidden states, D is the maximum residential time (or maximum duration, the maximum steps allowed to be in one state without transition).

Using the idea of representation graphs, a new aspect of the previous result could be given, with a similar, but new model and with a similar, but new learning algorithm which has the same computational complexity as in Yu & Kobayashi [4].

Firstly a new, general definition of RT-HMM should be established with the usage of representation graphs.

Definition 7. *(Residential time HMM)*

Let $\theta = (\pi, A, \theta_o)$ is an HMM with M different hidden states. π is the starting probability, 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 is a duration distribution for the hidden state i represented with graph $G_i(\eta_i)$. T_i comes from a duration distribution family represented with G_i. Let

- *$D_i = n(G_i)$*
- *$e^i_{in} = e_{in}(G_i)$*
- *$e^i = e(G_i)$*
- *$e^i_{out} = e_{out}(G_i)$*
- *$r_i = r(G_i)$ starting node*
- *$s_i = s(G_i)$ ending node*

These are independent of the values of η_i.

The residential time HMM $(\tilde\pi, \tilde A, \tilde\theta_o)$ constructed from θ and $\{G_i(\eta_i)\}$ is the following (for $i = 1,\dots,M$):

- *hidden states: i_d for $d = 1,\dots,D_i$*
- *transition probabilities*
  - *$\tilde A(i_k, i_l) = p(i_l|i_k)$ for $k,l = 1,\dots,D_i$*
  - *$\tilde A(i_k, j_l) = p(j_l|i_k) = p(s_i|i_k)A_{ij}p(j_l|r_j)$ for $k = 1,\dots,D_i$ for $j = 1,\dots,D_j$ for $j \neq i$*
- *starting probabilities $\tilde\pi(i_1) = \pi(i)$*
- *observation model parameters $\tilde\theta_o(i_k) = \theta_o(i)$ for $k = 1,\dots,D_i$*

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

The built RT-HMM has two layers of representation: a lower-level representation with i_d, which forms a Markov chain, aware of exactly where we are, and a higher-level representation with $i \leftrightarrow \{i_1,\dots,i_{D_i}\}$, which corresponds to the original hidden states, now with the desired residential times.

The number of (non-zero) edges in a dense RT-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 RT-HMM could be upper-bounded by V (starting probabilities) + 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 $D \gg M$.

As we will show, with the number of (non-zero) edges we could easily measure the computational complexity of the parameter learning of a specific HMM.

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

Definition 8. *(Representation efficiency of RT-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 RT-HMM:

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

Because we want to work with any HMM and the representation mostly relies on representing T_i with G_i, we could only observe the representation efficiency of the complete graphs.

Definition 9. *(Representation efficiency function)*

Consider the complete M graph as the Markov chain of an HMM. 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:

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

Now we can measure the goodness of representations together. Next, we want to measure the efficiency of individual representations. The motivation is the following: each T_i may come from the same family, and it simplifies the following thoughts. 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 10. *(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 edges number: $e_{in} = 1$, $e_{out} = 1$, $e = 1$. 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.

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 [4].

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 $D \geq 2$.

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 8. *(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 represent this family. Then

1) *G has no circle*
2) *G has at least D nodes (besides r and s)*
3) *$E(G, \{T(\theta)\})(M) \geq M(D-1) + M(M-1)$*

Thus, the second representation has efficiency $\mathcal{O}(MD + M^2)$ and the optimal efficiency is also $\mathcal{O}(MD + M^2)$.

VI. Learning parameters of RT-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 updated.

As the model is still an HMM, the E-step and every related computation could be done as before: $\alpha, \beta, \gamma, \xi$. Also, the Viterbi decoding could be done as before as well.

However, the M-step must be changed, because, from the definition of RT-HMM, some parameters are tied between states, therefore no individual update on states is allowed.

The starting probability update in M-step could be done as before, or with zeroing on $\{\pi(i_d) : d > 1\}$.

The observation model parameters are tied for every i, therefore one must sum up the statistics from across $\{i_1, \ldots, i_{D_i}\}$, then calculate an overall $\theta_o(i)$, finally assigns this parameter to every state: $\theta_o(i_d) = \theta_o(i)$.

The update on the transition model parameters is the most tricky, one must use the factorization $p(j_l|i_k) = p(s_i|i_k)A_{ij}p(j_l|r_j)$ in the log form: $\log p(j_l|i_k) = \log p(s_i|i_k) + \log A_{ij} + \log p(j_l|r_j)$, and group the members in the maximization of A to gain an analytical update. Here the "parameters" learned cannot be used directly in the model, one must use again the factorization to get $\hat{p}(j_l|i_k) = \hat{p}(s_i|i_k)\hat{A}_{ij}\hat{p}(j_l|r_j)$.

So, the reformulation of EM must be done, but (of course) it could be used for learning. On the computational complexity: in the simple HMM the E-step is $\mathcal{O}(M^2T)$, and the M-step is $\mathcal{O}(T\#\{\text{parameters}\}) = \mathcal{O}(T(M + M^2 + ML))$, where L is the number of parameters for $p(x_t|z_t = i)$.

In the RT-HMM case, we must revisit the computational complexity.

It is not hard to see, that the E-step could be done in $\mathcal{O}(TE(\{G_i\}, \{T_i\}, \theta)) = \mathcal{O}(T\#\{\text{non-zero edges}\}) = \mathcal{O}(TE)$, and the M-step could be done in $\mathcal{O}(T\#\{\text{parameters}\}) = \mathcal{O}(VL + E)$.

Applying this to the optimal representation of categorical distributions, we get back the result from Yu & Kobayashi, the step-size of the forward-backward algorithm (and of the E-step) is $\mathcal{O}(T(MD+M^2))$, and there is no faster way of this computation using representation graphs. The M-step could be done in $\mathcal{O}(T(M^2+MDL))$ steps, where it could be assumed that L is a small constant.

In the reasoning, we used that if in the Baum-Welch algorithm, we initialize A_{ij} with 0, then it does not change during the algorithm. Also the related $\xi_{t-1,t}(i,j)$ marginals would be 0 at every step, therefore we are allowed to skip the fix-zero edges in the learning steps.

References

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

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

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

[4] 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.