IFT 6269: Probabilistic Graphical Models **Fall 2018**

Lecture 15 — October 20

Lecturer: Simon Lacoste-Julien *Scribe: Samuel Beland-Leblanc*

Disclaimer: Lightly proofread and quickly corrected by Simon Lacoste-Julien.

15.1 HMM: Hidden Markov Model

The Hidden Markov Model (HMM) is a generalization of the latent variable model (such as the Gaussian mixture model GMM for example) with an added time dependence on the latent variables Z_t.

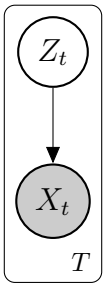

Figure 15.1: Latent variable model (GMM is an example)

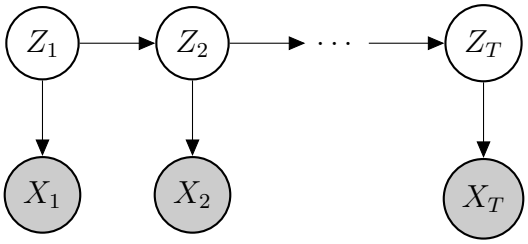

Figure 15.2: Latent variable model with added dependence on $Z_t \Rightarrow$ HMM

- Latent variable: $Z_t \in \{1, \ldots, k\}$, discrete
  - Later, $Z_t \sim$ Gaussian $\rightarrow$ Kalman Filter
- Observed variable: X_t
  - Continous (e.g. speech signal)
  - Discrete (e.g. DNA sequence)

From DGM theory, we get the following joint probability:

$$p(x_{1:T}, z_{1:T}) = p(z_1) \prod_{t=1}^{T} \underbrace{p(x_t|z_t)}_{\text{emission prob.}} \prod_{t=2}^{T} \underbrace{p(z_t|z_{t-1})}_{\text{transition prob.}} \tag{15.1}$$

Often, the **emissision probabilities** and the **transition probabilities** are homogeneous (i.e. they don't depend on t). Hence, we have that:

- $p_t(x_t|z_t) = f(x_t|z_t)$
- $p_t(z_t = i|z_{t-1} = j) = A_{ij}$
  - A is named the *Transition Matrix* (or *Stochastic Matrix*)
  - $\sum_i A_{ij} = 1, \forall j$. A column j of the transition matrix can be seen as a probability distribution over z_t.[1]

15.1.1 Inference Tasks

There are multiple inference tasks of interest when using HMM's. The general task is to compute the probability of a sequence of hidden state z given an observable output sequence x. But, there are also some marginal probabilities that are interesting to get :

- Prediction: $p(z_t|x_{1:t-1}) \rightarrow$ *Where next?*
- Filtering: $p(z_t|x_{1:t}) \rightarrow$ *Where now?*
  - The term *filtering* comes from the interpretation that the output x_t provides "noisy" information about the underlying "signal" z_t. So the "noisy" signals are filtered based on the value of $p(z_t|x_{1:t})$.
- Smoothing: $p(z_t|x_{1:T}), t < T \rightarrow$ *Where in the past?*

In order to perform these inferences, we need to take advantage of the conditional independence involved in the graphical model when conditioning on a latent variables. By conditioning on z_t, we make z_{t-1} independent of z_{t+1} (i.e. the future is independent of the past given the present). This thus gives us the following :

[1] Note that some textbooks use a normalized row convention instead of our normalized column one. Simon prefers the column convention as then the updates are matrix vector products (see the HMM message passing updates later).

$$\begin{aligned} p(z_t|x_{1:T}) &= \frac{p(x_{1:T}|z_t)p(z_t)}{p(x_{1:T})} \\ &= \frac{p(x_{1:t}|z_t)p(x_{t+1:T}|z_t)p(z_t)}{p(x_{1:T})} \\ &= \frac{p(x_{1:t}, z_t)p(x_{t+1:T}|z_t)}{p(x_{1:T})} \\ &= \frac{\alpha(z_t)\beta(z_t)}{p(x_{1:T})} \\ &= \frac{\alpha(z_t)\beta(z_t)}{\sum_{z_t}\alpha(z_t)\beta(z_t)} \end{aligned}$$

Where α and β are two recursion that we will define.

α-recusion

We will use the sum product algorithm here to derive recursions to compute the probabilities (as a didactic example of sum product on UGMs – one can also derive these recursions directly instead).

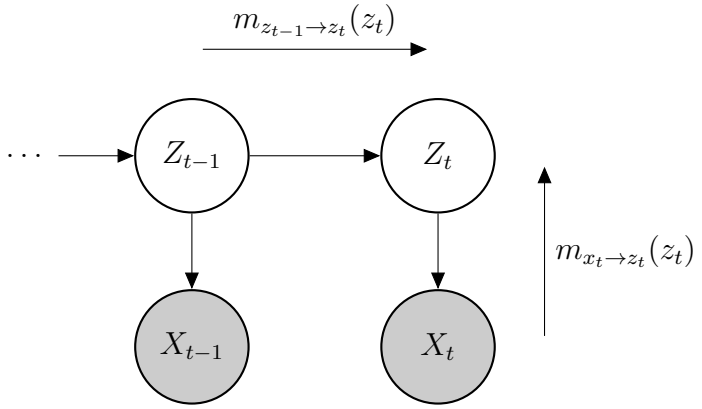

Figure 15.3: Visual representation of α-recursion

Instead of computing the filtering distribution, we will compute the joint marginal $p(z_t, \bar{x}_{1:t}) \propto p(z_t|\bar{x}_{1:t})$ using message passing. Here we are using the $\bar{x}$ notation to indicate that the observation are **fixed** for the marginalization. So we get:

$$
\begin{aligned}
p(z_t, \bar{x}_{1:t}) &= \frac{1}{Z} 1 \cdot m_{z_{t-1} \to z_t}(z_t) \cdot m_{x_t \to z_t}(z_t) \\
\text{with } m_{x_t \to z_t}(z_t) &= \sum_{x_t} p(x_t|z_t)\delta(x_t, \bar{x}_t) = p(\bar{x}_t|z_t) \\
\text{with } m_{z_{t-1} \to z_t}(z_t) &= \sum_{z_{t-1}} p(z_t|z_{t-1}) \underbrace{m_{z_{t-2} \to z_{t-1}}(z_{t-1}) \cdot m_{x_{t-1} \to z_{t-1}}(z_{t-1})}_{p(z_{t-1}, \bar{x}_{1:t-1}) = \alpha_{t-1}(z_{t-1})}
\end{aligned}
$$

Note that $Z = 1$ above as we had a DGM; and the 1 in the first equation is because we did not have any node potential.

Let's then define : $\alpha_t(z_t) \triangleq p(z_t, \bar{x}_{1:t})$, which can be expressed using the above derivations (making the recursion explicit) as :

$$\boxed{\alpha_t(z_t) = p(\bar{x}_t|z_t) \sum_{z_{t-1}} p(z_t|z_{t-1})\alpha_{t-1}(z_{t-1})} \tag{15.2}$$

This is the α-recursion (a.k.a *forward recursion*). It is like the *collect phase* in the sum product algorithm using z_t as the root. We can also express it as a matrix-vector product. From the definition we just proposed, we can see that :

$$\alpha_t(z_t) = \underbrace{p(\bar{x}_t|z_t)}_{\text{vector}(z_t)} \sum_{z_{t-1}} \underbrace{p(z_t|z_{t-1})}_{\text{matrix}} \underbrace{\alpha_{t-1}(z_{t-1})}_{\text{vector}}$$

Let $O_t(z_t) \triangleq p(\bar{x}_t|z_t)$, then using the *Hadamard product* ($\odot$) we can redefine the α-recusion like this:

$$\boxed{\alpha_t = O_t \odot A\alpha_{t-1}} \tag{15.3}$$

The initialization for the α-recursion is simply : $\alpha_1(z_1) = p(z_1, \bar{x}_1) = p(z_1)p(\bar{x}_1|z_1)$. Also, we can observe that if we renormalize α_t over z_t, we get our filtering distribution $\tilde{\alpha}_t \triangleq p(z_t|\bar{x}_{1:t})$. From the α, we can also get the *evidence probability*:

$$\sum_{z_t} p(z_t, \bar{x}_{1:t}) = \sum_{z_t} \alpha_t(z_t) = p(\bar{x}_{1:t}) \tag{15.4}$$

Time complexity: $\mathcal{O}(t \cdot k^2)$ (k^2 for the matrix/vector products over k states repeated t times)

Space Complexity: We only need an extra storage of $\mathcal{O}(k)$ for the alpha recursion. Note that it takes $\mathcal{O}(k^2)$ to store the whole A matrix (i.e. *transition matrix*), but this is given by the problem, so it is not "extra storage".

β-recursion : smoothing

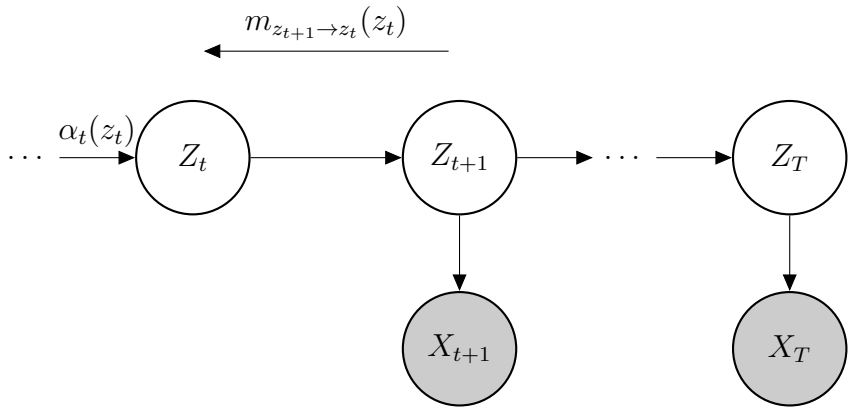

Figure 15.4: Visual representation of the β-recursion

To get our smoothing probability, we need to also consider the information for $T > t$; this is where the beta recursion is needed. To get the joint marginal on z_t and all the observations, we have :

$$p(z_t, \bar{x}_{1:T}) = \frac{1}{Z}\alpha_t(z_t) \cdot \underbrace{m_{z_{t+1}\to z_t}(z_t)}_{\triangleq \beta_t(z_t)} \tag{15.5}$$

From the conditional independence property we explained earlier, we get :

$$\boxed{\beta_t(z_t) \triangleq p(\bar{x}_{t+1:T}|z_t)} \tag{15.6}$$

By expanding the message in equation 15.5, we can expose the actual recursion :

$$m_{z_{t+1}\to z_t}(z_t) = \sum_{z_{t+1}} p(z_{t+1}|z_t)p(\bar{x}_{t+1}|z_{t+1})m_{z_{t+2}\to z_{t+1}}(z_{t+1}) \tag{15.7}$$

$$\boxed{\beta_t(z_t) = \sum_{z_{t+1}} p(z_{t+1}|z_t)p(\bar{x}_{t+1}|z_{t+1})\beta_{t+1}(z_{t+1})} \tag{15.8}$$

With the following initialization : $\beta_T(z_T) = 1, \forall z_T$.[2]
Finally, from the sum-product algorithm, we can obtain the **edge marginal** as :

$$p(z_t, z_{t+1}, \bar{x}_{1:T}) = \alpha_t(z_t)\beta_{t+1}(z_{t+1})p(z_{t+1}|z_t)p(\bar{x}_{t+1}|z_{t+1}) \tag{15.9}$$

[2]This can be seen as we do not observe anything for $t > T$, so marginalizing all the leaves of a DGM there just yields the value 1. (Leaf plucking property)

15.1.2 Numerical Stability Trick

A big problem with doing inference in HMM's is the amount of multiplication of values $<< 1$, which makes it so that α_t and β_t can easily go to $1e-100$. This is bad as it can underflow. There are 2 tricks that can be used in order to avoid this.

(A) (General) Store $\log(\alpha_t)$ instead

Let:

$$\tilde{a} \triangleq \max_i a_i$$
$$i_{\max} \triangleq \arg\max_i a_i$$

Then we use the following :

$$\begin{aligned}\log\left(\sum_i a_i\right) &= \log\left(\tilde{a}\left(\sum_i \frac{a_i}{\tilde{a}}\right)\right) \\ &= \log(\tilde{a}) + \log\left(1 + \sum_{j \neq i_{\max}} \exp\left(\log(a_i) - \log(\tilde{a})\right)\right)\end{aligned}$$

(B) Normalize the Messages

For the **α-recursion**. we can use our previously defined $\tilde{\alpha}_t(z_t) = p(z_t|\bar{x}_{1:t})$ (filtering distribution). We initially had $\alpha_t(z_t) = O_t(z_t) \odot A\alpha_{t+1}(z_{t-1})$. Now, we get:

$$\boxed{\tilde{\alpha}_t = \frac{O_t(z_t) \odot A\tilde{\alpha}_{t+1}(z_{t-1})}{\sum_{z_t}(O_t(z_t) \odot A\tilde{\alpha}_{t+1}(z_{t-1}))}} \tag{15.10}$$

It is possible to show that:

$$\begin{aligned}\sum_{z_t}(O_t(z_t) \odot A\tilde{\alpha}_{t+1}(z_{t-1})) &= p(\bar{x}_t|\bar{x}_{1:t-1}) \\ &\triangleq c_t\end{aligned}$$

We hence get : $p(\bar{x}_{1:T}) = \prod_{t=1}^T p(\bar{x}_t|\bar{x}_{1:t-1}) = \prod_{t=1}^T c_t$.
Now, for the **β-recursion**, we define :

$$\tilde{\beta}(z_t) \triangleq \frac{\beta_t(z_t)}{p(\bar{x}_{t+1:T}|\bar{x}_{1:t})} = \frac{\beta_t(z_t)}{\prod_{u=t+1}^T c_u} \tag{15.11}$$

Note here that $\sum_{z_t} \tilde{\beta}(z_t) \neq 1$ in general, but it will have a reasonable value (not underflow), and has the advantage of not requiring much extra computation by re-using the stored c_t values. Exercise: derive the $\tilde{\beta}$-recursion.

15.1.3 Maximum Likelihood for HMM

First of all, let:

$$\begin{aligned} p(x_t|z_t=k) &= f(x_t|\eta_k), \eta=(\eta_k)_{k=1}^K && \text{(for some parametric model (e.g. Gaussian))} \\ p(z_{t+1}=i|z_t=j) &= A_{i,j}, && \text{(where } A \text{ is the transition matrix)} \\ p(z_1=i) &= \pi_i && \text{(since } z_1 \text{ has no parents)} \end{aligned}$$

We want to estimate our parameters $\hat{\theta}=\{\hat{\eta},\hat{A},\hat{\pi}\}$ from the sequences of data $(x^{(i)})_{i=1}^N$, where $x^{(i)}=x_{1:T_i}^{(i)}$. As we have a latent variable model, we are going to use **EM**.

E-step

Let s be the s^{th} iteration. Then our **E-step** at time $s+1$ is simply our $\alpha-\beta$ recursion with our parameters at time s:

$$\boxed{q_{s+1}=p(z|x,\theta^{(s)})} \tag{15.12}$$

M-step

We are trying to optimize :

$$\hat{\theta}^{(s+1)}=\arg\max_{\theta\in\text{(H)}} \mathbb{E}_{q_{s+1}}[\log p(x,z)] \tag{15.13}$$

For this we are going to use the complete log-likelihood:

$$\log p(x,z|\theta)=\sum_{i=1}^N\left[\log p(z_1^{(i)})+\sum_{t=1}^T\log p(\bar{x}_t^{(i)}|z_t^{(i)})+\sum_{t=2}^T\log p(z_t^{(i)}|z_{t-1}^{(i)})\right] \tag{15.14}$$

Now if we look at each term individually, we will be able to maximize with respect to θ after.

1. $\log p(z_1^{(i)}) \Rightarrow \sum_k z_{1,k}^{(i)}\log\pi_k$
2. $\log p(\bar{x}_t^{(i)}|z_t^{(i)}) \Rightarrow \sum_k z_{t,k}^{(i)}\log f(\bar{x}_t^{(i)}|\eta_k)$
   - $\mathbb{E}_{q_{s+1}}\left[z_{t,k}^{(i)}\right]=q_{s+1}(z_{t,k}^{(i)}=1)\triangleq\tau_{t,k}^{(i)}$ (soft counts)
   - $q_{s+1}(z_{t,k}^{(i)}=1)$ is our smoothing distribution $p(z_t^{(i)}|\bar{x}_{1:T_i}^{(i)})$
3. $\log p(z_t^{(i)}|z_{t-1}^{(i)}) \Rightarrow \sum_{l,m} z_{t,l}^{(i)}z_{t-1,m}^{(i)}\log A_{l,m}$
   - $z_{t,l}^{(i)}z_{t-1,m}^{(i)} \Rightarrow q_{s+1}(z_{t,l}^{(i)}=1,z_{t-1,m}^{(i)}=1)\triangleq\tau_{t,l,m}^{(i)}$ (soft counts)
   - $q_{s+1}(z_{t,l}^{(i)}=1,z_{t-1,m}^{(i)}=1)$ is our smoothing edge marginal $p(z_{t,l}^{(i)}=1,z_{t-1,m}^{(i)}=1|\bar{x}_{1:T_i}^{(i)},\theta^{(s)})$

Maximize with respect to θ

$$\hat{\pi}_k^{s+1} = \frac{\sum_{i=1}^N \tau_{1,k}^{(i)}}{\sum_{i=1}^N \underbrace{\sum_{l=1} \tau_{1,l}^{(i)}}_{1}} = \frac{\sum_{i=1}^N \tau_{1,k}^{(i)}}{N} \tag{15.15}$$

$$\hat{A}_{l,m}^{(s+1)} = \frac{\sum_{i=1}^N \sum_{t=2}^T \tau_{t,l,m}^{(i)}}{\sum_u \sum_{i=1}^N \sum_{t=2}^T \tau_{t,u,m}^{(i)}} \tag{15.16}$$

As for $\hat{\eta}_k$, this will depend on the parametric model used, but you get them using soft count maximum likelihood similar to how it was used in GMM (e.g. for Gaussians we had the *weighted empirical mean*)

We just described what is called the **Baum-Welch** algorithm consisting of forward-backward using $\alpha - \beta$ recursion/sum-product with **EM** for HMM's.

Finally, to find $\arg\max_{z_1,\ldots,z_T} p(z_{1:T}|\bar{x}_{1:T})$, we must use the **Viterbi** algorithm (i.e. max product) seen earlier.