Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.S064 Introduction to Machine Learning

Phase 3: Hidden Markov Models cont'd (lecture 17)

In this lecture, we will explore the connection between two classes of generative models that we have studied in previous lectures: mixture models and HMMs. Both of these models involve inferring values for unobserved (latent) variables given values for "observed" variables. In mixture models, latent variables y are components or clusters (also class labels) while the observations x are vectors or symbols. In the context of HMMs, latent variables y are "tags" or "states" while observations x are output symbols such as words or (more generally) vectors.

Our goal is to highlight similarities and differences between these models in light of their inherent "modeling assumptions". These assumptions have consequences in terms of what the models can capture, how the latent variables are inferred, or how we can estimate these models from incomplete data. For this purpose, we will look at fixed length n sequences

$$y_1, y_2, \ldots, y_n$$
$$x_1, x_2, \ldots, x_n$$

where, typically, variables y_i, $i = 1, \ldots, n$, are not observed while words x_i are. Both of our models can be used in this setting.

Independent Pair Generation We first assume that all the words in the sequence are generated independently from the same mixture model. The mixture model is defined by the prior (tag) probabilities $P(y)$ together with the corresponding tag-conditional output distributions $P(x|y)$. The model assigns probability

$$P(x_i|y_i)P(y_i)$$

to any observed pair (x_i, y_i), reflecting the fact that y_i is sampled from $P(y)$ and x_i is subsequently sampled from $P(x|x_i)$. Now, since each pair in the sequence is assumed to be independent of other pairs, the probability of the whole sequence of pairs is:

$$\begin{aligned} P((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)) &= P(x_1|y_1)p(y_1)P(x_2|y_2)P(y_2)\ldots P(x_n|y_n)P(y_n) \quad (1) \\ &= \prod_{i=1}^{n} [P(y_i)P(x_i|y_i)] \quad (2) \end{aligned}$$

Let's now think about the case where y's are not observed. The marginal probability of any observation x is given by

$$P_x(x) = \sum_y P(x|y)P(y) \tag{3}$$

where we are summing over all the ways that x could be generated from y. Since we assume that each (x_i, y_i) pair in the sequence was independent of others, clearly

$$P(x_1, \ldots, x_n) = \prod_{i=1}^n P_x(x_i) = \prod_{i=1}^n \left[\sum_{y_i} P(x_i|y_i)P(y_i)\right] \tag{4}$$

We should obtain the same answer by summing over the y_i's in Eq.(2). Indeed we do

$$\begin{aligned} P(x_1, \ldots, x_n) &= \sum_{y_1,\ldots,y_n} P((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)) & (5) \\ &= \sum_{y_1,\ldots,y_n} \prod_{i=1}^n [P(x_i|y_i)P(y_i)] = \prod_{i=1}^n \left[\sum_{y_i} P(x_i|y_i)P(y_i)\right] & (6) \end{aligned}$$

Can we think of this model as an HMM? In order to interpret the mixture model as an HMM, we would have to set the prior probabilities, transition probabilities, and the emission probabilities in such a way that the HMM would specify exactly the same joint distribution over words x_i and tags y_i, i.e.,

$$P_{HMM}(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n) = \prod_{i=1}^n P(x_i|y_i)P(y_i)$$

We can indeed achieve this if the HMM parameters are defined as follows:

$$\begin{aligned} b_y(x) &= P(x|y) & (7) \\ a_{ij} &= P(y = j) & (8) \\ \pi_i &= P(y_1 = i) & (9) \end{aligned}$$

Note that the transition probabilities in this case do not depend on the current tag/state at all. In fact, regardless of the current state, the next state is chosen with probability $P(y_i)$, the prior probability in the mixture model. As a result, there's no dependence between successive states. The corresponding trellis for the cases of two possible labels y is shown below (note that the independence is hidden in the values of the transitions rather than in the structure of the trellis).

Let's now review how we estimate the mixture model when y's are not observed. We assume the same mixture for all the documents, and for all the words in the document. We aim to maximize $P(x_1, x_2, \ldots, x_n)$ (for simplicity, restricting the estimation here for

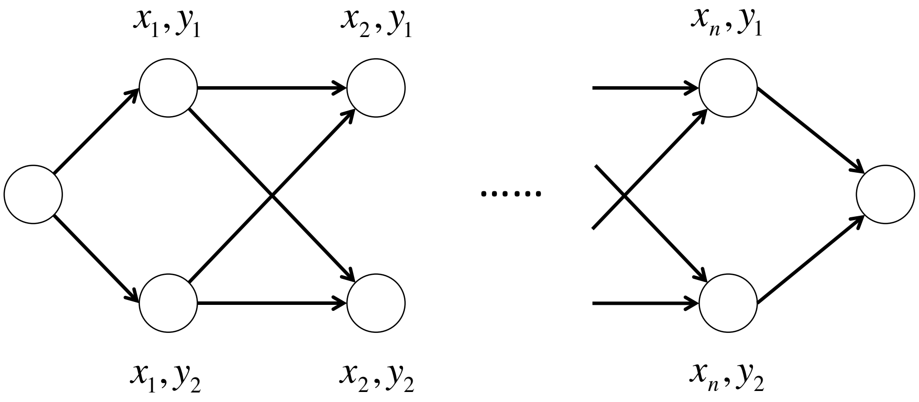

Figure 1: The corresponding trellis.

a single document). Our goal is to find $P(y)$ and $P(x|y)$ that maximize $P(x_1, \ldots, x_n)$. This is equivalent to maximizing the log-likelihood:

$$\log P(x_1, x_2, \ldots, x_n) = \log \prod_{i=1}^{n} P_x(x_i) = \sum_{i=1}^{n} \log P_x(x_i) \tag{10}$$

Let's write an EM algorithm for this model:

E-step

If we had tags $y_1, \ldots, y_n$, we would evaluate the necessary counts as

$$\text{count}(x, y) = \sum_{i=1}^{n} [\![\, x = x_i \;\&\; y = y_i \,]\!] \tag{11}$$

where, as before, $[\![\cdot]\!]$ is an indicator function of the statement inside. Since the tags are not given, we must infer them. Using the current model parameters, we evaluate the posterior probability of y_i given the observed word x_i:

$$P(y_i|x_i) = \frac{P(x_i|y_i)P(y_i)}{\sum_y P(x_i|y)P(y)} \tag{12}$$

Based on these posterior probabilities, the expected counts become

$$\begin{aligned}
\overline{\text{count}}(x,y) &= E\{\sum_{i=1}^{n}[\![\, x = x_i \,\&\, y = y_i \,]\!]|x_1,\ldots,x_n\} && (13)\\
&= \sum_{i=1}^{n} E\{[\![\, x = x_i \,\&\, y = y_i \,]\!]|x_i\} && (14)\\
&= \sum_{i=1}^{n}\sum_{y_i} P(y_i|x_i)[\![\, x = x_i \,\&\, y = y_i \,]\!] && (15)\\
&= \sum_{i=1}^{n} P(y|x_i)[\![\, x = x_i \,]\!] && (16)
\end{aligned}$$

M-step

$$\begin{aligned}
\hat{P}(y) &= \frac{\sum_x \overline{\text{count}}(x,y)}{\sum_{x,y'} \overline{\text{count}}(x,y')} && (17)\\
\hat{P}(x|y) &= \frac{\overline{\text{count}}(x,y)}{\sum_{x'} \overline{\text{count}}(x',y))} && (18)
\end{aligned}$$

Single Tag Sequence Generation Let's consider a model at the other extreme. Instead of generating all the tags independently, we assume that the words in the document (sequence) share the same tag. In other words, we generate the tag only once. The only possible pairs of sequences are then

$$\begin{array}{cccc} y & y & \ldots & y \\ x_1 & x_2 & \ldots & x_n \end{array}$$

The joint distribution over tags and the words is then

$$P(x_1,\ldots,x_n,y) = P(y)\prod P(x_i|y) \tag{19}$$

Note that $P(x_i|y)$ have the same form as before (but they will not be the same as before after estimation). The probability of the whole sequence is also different since the shared tag y is generated only once.

When y is unknown, then any individual x_i has the marginal probability given by

$$P_x(x_i) = \sum_y P(x_i|y)P(y) \tag{20}$$

This looks exactly the same as before. However, since the tag is shared, the marginal probability over all the x's is quite different

$$P(x_1,\ldots,x_n) = \sum_y P(y)\prod_i P(x_i|y) \tag{21}$$

Is it possible to view this model as an HMM as well? Yes, it is. Again, we have to specify the HMM parameters in such a way that the joint distribution Eq.(19) agrees with that of the HMM. This holds when

$$b_y(x) = P(x|y) \tag{22}$$

$$a_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

$$\pi_i = P(y_1 = i) \tag{24}$$

In other words, y_1 is the shared tag, sampled from $P(y)$ as it should. The transition probabilities ensure that the tag does not change and all the observations are generated using the same tag. The trellis constructed for this HMM now looks like:

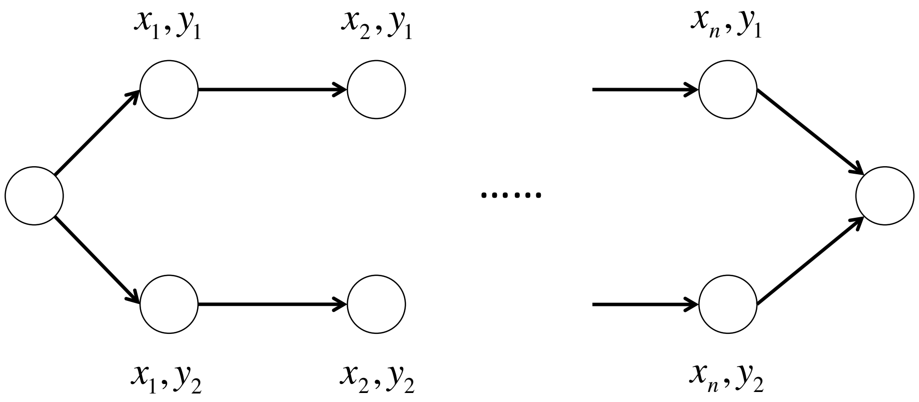

Figure 2: The corresponding trellis.

Let's see how the EM algorithm looks for this model.

E-step

Since the tag is not known, we must infer it from the observations

$$P(y|x_1 \dots x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i|y)}{P(x_1 \dots x_n)} = \frac{P(y) \prod_{i=1}^{n} P(x_i|y)}{\sum_{y'} P(y') \prod_{i=1}^{n} P(x_i|y')} \tag{25}$$

Unlike before, the posterior depends on all the $x_1, \dots, x_n$. This is expected since the tag is shared (each x_i provides some evidence about what the underlying y should be). The expected counts are then

$$\overline{\text{count}}(x, y) = \sum_{i=1}^{n} P(y|x_1 \dots x_n)[\![x_i = x]\!] \tag{26}$$

$$= P(y|x_1 \dots x_n) \sum_{i=1}^{n} [\![x_i = x]\!] \tag{27}$$

$$= P(y|x_1 \dots x_n) \text{ count}(x) \tag{28}$$

M-step The M-step is the same as for the previous model.

General HMM After these two special cases of dependencies along the sequence (no dependence, shared tag), we can look at the more general case where each tag depends only on the tag that came before.

$$P(x_1, \ldots, x_n, y_1, \ldots, y_n) = P(y_1) \prod_{i=1}^{n} P(x_i|y_i) \prod_{i=2}^{n} P(y_i|y_{i-1}) \tag{29}$$

As before, we are interested in predicting y given the observations x. Given that y_i's are intertwined in a particular way, we need an algorithm that can handle both the cases of independent tags (the first model) and fully dependent (the second model). There are two relevant sources of information when we predict each tag:

- Previous words in the sequence as their tags influence the current one
- Future words in the sequence as their underlying tags depend on the current one

The goal is to express these sources of information by summarizing the past and the future. For this reason, we introduce forward and backward probabilities (also discussed in previous lecture). We will take a closer look at the forward probability here. There are many ways to define the forward probabilities. We will adopt here the convention that the forward probabilities specify the likelihood of generating observations (up to but not including the current one) with the constraint that we end up with a particular tag at the current step.

$$\alpha_y(j) = P(x_1, \ldots, x_{j-1}, y_j = y) = \sum_{y_1, \ldots, y_{j-1}} P(x_1, \ldots, x_{j-1}, y_1, \ldots, y_{j-1}, y_j = y) \tag{30}$$

$$\begin{array}{ccccc} ? & ? & \ldots & ? & y \\ x_1 & x_2 & \ldots & x_{j-1} & ? \end{array}$$

where the question marks are placeholders for the unobserved variables. Since we need to sum over all the possible values for $y_1 \ldots y_{j-1}$ to obtain the forward probabilities, it may appear that the computation is prohibitively costly. We will show below how this computation can be done using dynamic programming. Note also that if we do have the forward probabilities, we can easily predict the next tag along the sequence:

$$P(y_j = y|x_1, \ldots, x_{j-1}) = \frac{\alpha_y(j)}{\sum_{y'} \alpha_{y'}(j)} \tag{31}$$

The forward algorithm (dynamic programming)

- **Base Case:** We have no observations and aim to predict the current (first) tag. This is given by the prior tag probability

$$\alpha_y(1) = \pi(y) \tag{32}$$

- **Recursive Case:** Here we assume that we have already evaluated $\alpha_{y'}(j-1)$ for all y'. In order to extend these to $\alpha_y(j)$ we have to generate the observation at step $j-1$ and transition into tag y at step j. Since the value of y' at step $j-1$ is unknown, we will sum over it, i.e., we consider all the possible ways of generating the observations, and transitioning into y at step j:

$$\alpha_y(j) = \sum_{y'} \alpha_{y'}(j-1) b_{y'}(x_{j-1}) a_{y'y} \tag{33}$$

Now let's see if this definition of forward probability α makes sense in our previous models. For illustrative purposes, we will look at sequences of length two.

Independent Pair Generation In our first model, we assumed that all the pairs are independent. Clearly

$$\begin{aligned}
\alpha_y(1) &= P(y_1 = y) && (34)\\
\alpha_y(2) &= \sum_{y'} P(y_1 = y')P(x_1|y_1 = y')P(y_2 = y) && (35)\\
&= P(y_2 = y) \underbrace{\sum_{y'} P(y_1 = y')P(x_1|y_1 = y')}_{\text{doesn't depend on } y} && (36)
\end{aligned}$$

so

$$P(y_2 = y|x_1) = \frac{\alpha_y(2)}{\sum_{y'} \alpha_{y'}(2)} = P(y_2 = y) \tag{37}$$

as it should since, when each pair is independent of others, there's no impact from an earlier observation.

Single Tag Sequence Generation In our second model, we assumed that all the words in the sequence share the same tag. In this case,

$$\begin{aligned}
\alpha_y(1) &= P(y_1 = y) && (38)\\
\alpha_y(2) &= \sum_{y'} \alpha_{y'}(1) b_{y'}(x_1) a_{y'y} = P(y_1 = y)P(x_1|y_1 = y) && (39)
\end{aligned}$$

so

$$P(y_2 = y|x_1) = \frac{P(y_1 = y)P(x_1|y_1 = y)}{\sum_{y'} P(y_1 = y')P(x_1|y_1 = y')} \tag{40}$$

as it should since x_1 does provide information about the shared tag.