Recitation 8
Hidden Markov Models and Bayes Net

10-601: Introduction to Machine Learning

11/3/2021

1 HMMs

You are given the following training data:

```
win_C league_C Liverpool_D

win_C Liverpool_D league_C

Liverpool_D win_C
```

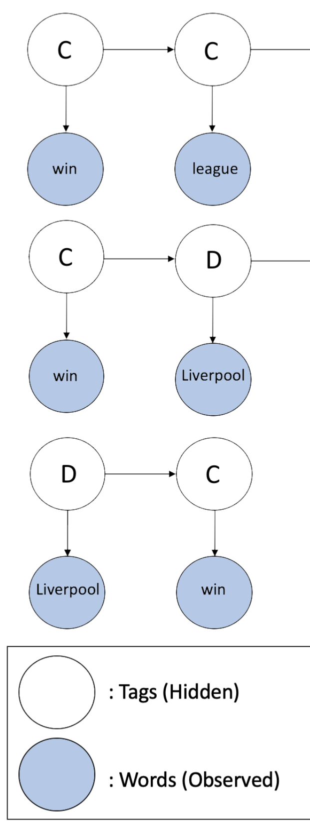

Figure 1: Visualization of Sequences

You are also given the following observed (validation) data: `Liverpool win league`

In this question, let each observed state $x_t \in \{1, 2, 3\}$, where 1 corresponds to `win`, 2 corresponds to `league`, and 3 corresponds to `Liverpool`. Let each hidden state $Y_t \in \{C, D\}$, where $s_1 = C$ and $s_2 = D$.

1. First, we need to train our HMM by generating the initial probabilities: $\boldsymbol{\pi}$, the transition probability matrix: $\mathbf{B}$, the emission probability matrix: $\mathbf{A}$.

   (a) Find $\boldsymbol{\pi}$. Recall that $\pi_j = P(Y_1 = s_j)$.

   (b) Find Transition Matrix: $\mathbf{B}$. Recall that $B_{jk} = P(Y_t = s_k \mid Y_{t-1} = s_j)$

   (c) Find Emission Matrix: $\mathbf{A}$. Recall that $A_{jk} = P(X_t = k \mid Y_t = s_j)$.

2. What is the likelihood of observing this output?
Recall that:

$$\alpha_t(k) = P(x_{1:t}, Y_t = s_k)$$

$$\beta_t(k) = P(x_{t+1:T} | Y_t = s_k)$$

We also have the recursive procedure:

(a) $\alpha_1(j) = \pi_j A_{jx_1}$.

(b) For $t > 1$, $\alpha_t(j) = A_{jx_t} \sum_{k=1}^{J} \alpha_{t-1}(k) B_{kj}$

You are now told that the observed data has the following tags:

```
Liverpool_D win_C league_D
```

3. Given the observed sequence of words (denote $\vec{x} = [\text{Liverpool, win, league}]^T$), what is the probability of these assigned tags $P(Y_1 = D|\vec{x}), P(Y_2 = C|\vec{x}), P(Y_3 = D|\vec{x})$?

   Recall that:

   $$P(Y_t = s_k|\vec{x}) = \frac{\alpha_t(s_k)\beta_t(s_k)}{P(\vec{x})}$$

   So, we need to find β_T

   We also have a similar recursive procedure

   (a) $\beta_T(j) = 1$ (All states could be ending states)

   (b) For $1 \leq t \leq T-1$, $\beta_t(j) = \sum_{k=1}^{J} A_{kx_{t+1}}\beta_{t+1}(k)B_{jk}$ (Generate x_{t+1} from any state)

4. The sequence of words you observe is again the same: `Liverpool win league`

   However, you are only given the tag of the last word: `league_C`

   Using the Viterbi Algorithm, what is the most likely sequence of hidden states?

   Recall that:

$$\omega_t(s_k) = \max_{y_{1:t-1}} P(x_{1:t}, y_{1:t-1}, y_t = s_k)$$

$$b_t(s_k) = \arg\max_{y_{1:t-1}} P(x_{1:t}, y_{1:t-1}, y_t = s_k)$$

   Also, the recursive procedure for the Viterbi algorithm is as follows:

   (a) $\omega_0(s_k) = 1$ for $s_k =$ `START` and 0 for all other states.

   (b) For $t > 1$,

   - $\omega_t(s_j) = \max_{1 \le k \le J} \omega_{t-1}(s_k) P(x_t | Y_t = s_j) P(Y_t = s_j | Y_{t-1} = s_k)$
     $= \max_{1 \le k \le J} \omega_{t-1}(s_k) A_{jx_t} B_{kj}$
   - $b_t(s_j) = \arg\max_{1 \le k \le J} \omega_{t-1}(s_k) P(x_t | Y_t = s_j) P(Y_t = s_j | Y_{t-1} = s_k)$
     $= \arg\max_{1 \le k \le J} \omega_{t-1}(s_k) A_{jx_t} B_{kj}$

   What is the most likely sequence of tags given the observed data? (Select `C` if tie)

2 Working in Log-space

2.1 Motivation

Given the following series of probability values:

$P(x_1 = 1)$	$P(x_2 = 1 \mid x_1 = 1)$	$P(x_3 = 1 \mid x_2 = 1, x_1 = 1)$
0.002	0.004	0.003

We want to find $P(x_1 = 1, x_2 = 1, x_3 = 1)$. Suppose we have a calculator which only has 4 decimal places of precision, so it can only store values of format X.XXXX

1. What is the correct value of $P(x_1 = 1, x_2 = 1, x_3 = 1)$ without any precision limits?

   $P(x_1 = 1, x_2 = 1, x_3 = 1) =$

2. What is the value of $P(x_1 = 1, x_2 = 1, x_3 = 1)$ using our faulty calculator?

   $P(x_1 = 1, x_2 = 1) =$

   $P(x_1 = 1, x_2 = 1, x_3 = 1) =$

3. How do the values of $P(x_1 = 1, x_2 = 1, x_3 = 1)$ from part (1) and (2) compare?

4. What is the value of $P(x_1 = 1, x_2 = 1, x_3 = 1)$ if we perform the same computation but in log space?

   $\log\big(P(x_1 = 1, x_2 = 1, x_3 = 1)\big) =$

   This is good! But we can use the log sum exp trick to extend its use to even smaller scales.

2.2 Forward and Backward Algorithm in Log Space

In the forward algorithm, recall that the α's can be computed using the recursive procedure:

- $\alpha_1(j) = \pi_j A_{jx_1}$
- For $t > 1$, $\alpha_t(j) = A_{jx_t} \sum_{k=1}^{J} \alpha_{t-1}(k) B_{kj}$

1. Derive $\log\big(\alpha_1(j)\big)$ in terms of $\log(\pi_j)$ and $\log(A_{jx_1})$
2. Derive $\log\big(\alpha_t(j)\big)$ in terms of $\log\big(\alpha_{t-1}(k)\big)$ and $\log A_{kj}$

In the backward algorithm, we also have a similar recursive procedure:

- $\beta_T(j) = 1$
- For $1 \leq t \leq T-1$, $\beta_t(j) = \sum_{k=1}^{J} A_{kx_{t+1}} \beta_{t+1}(k) B_{jk}$

1. Derive $\log\big(\beta_T(j)\big)$
2. Derive $\log\big(\beta_t(j)\big)$ in terms of $\log(A_{kx_{t+1}})$, $\log\big(\beta_{t+1}(k)\big)$, and $\log(B_{jk})$

3 Bayesian Networks

3.1 Practice problems

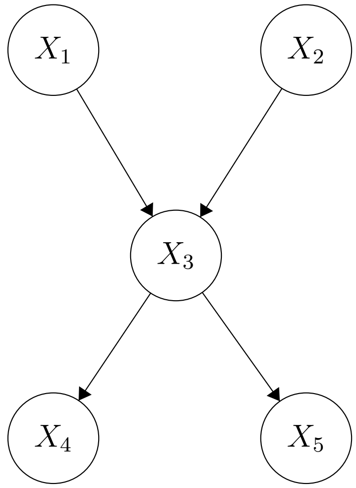

Figure 2: Graphical Model

1. Write down the factorization of the above directed graphical model.

   $P(X_1, X_2, X_3, X_4, X_5)$