ECE 417 Multimedia Signal Processing
Homework 4

UNIVERSITY OF ILLINOIS
Department of Electrical and Computer Engineering

Assigned: Tuesday, 10/8/2020; Due: Monday, 10/19/2020
Reading: L.R. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, 1989

Problem 4.1

Write a phonemic transcription of the sentence "At the still point, there the dance is" (by T.S. Eliot) using either IPA or ARPABET.

Solution: In ARPABET:

`AE T DH AX S T I H L P OI N T DH EH R DH AZ D AE N S I Z`

In IPA: æ t ð ə s t ɪ l p ɔɪ n t ð ɛ ɻ ð ə d æ n s ɪ z

Problem 4.2

The softmax computes an estimate of the state posterior pmf, $p(q|\vec{x})$. As discussed in lecture, you can't compute exactly the likelihood from the softmax, but you can compute it up to a constant factor $G[t]$:

$$b_q[t] = \frac{G[t]\exp(e_q[t])}{p(q)},$$

where $p(q) \in [0, 1]$ is the prior probability of q, $e_q[t]$ is the q^{th} node of the neural network's final-layer excitation in frame t, and $G[t]$ is a constant, in the sense that it depends on t, but not on q. $G[t]$ is unknown, but an estimate with nice numerical properties is

$$G[t] = \frac{1}{\max_j \exp(e_j[t])}$$

In HMM training with known segmentation, the parameters of the HMM might be trained using a kind of maximum-likelihood criterion similar to cross-entropy, specifically, the network parameters are trained to minimize

$$\mathcal{L} = -\sum_{i=1}^{N} \sum_{t:q_t=i} \ln b_i[t],$$

where you may assume that q_t, the state variable at time t, is known. Find $\frac{d\mathcal{L}}{de_q[\tau]}$, for some particular value of τ, for all values of q. Be careful:

- Notice that $G[\tau]$ depends on $e_j[\tau]$, even for values of j other than q_t.
- You may find it useful to consider, separately, the following four cases:

(a) $q = q_\tau$

(b) $q = \text{argmax}_j\, e_j[\tau]$

(c) Both of the above

(d) Neither of the above

Solution:

$$\frac{d\mathcal{L}}{de_q[\tau]} = -\frac{1}{b_{q_t}[\tau]}\left(\frac{G[t]}{p[q]}\frac{d\exp(e_{q_\tau}[\tau])}{de_q[\tau]} + \frac{\exp(e_{q_\tau}[\tau])}{p[q]}\frac{dG[\tau]}{de_q[\tau]}\right)$$

where

$$\frac{d\exp(e_{q_\tau}[\tau])}{de_q[\tau]} = \begin{cases} \exp(e_{q_\tau}[\tau]) & q_\tau = q \\ 0 & \text{otherwise} \end{cases}$$

and

$$\frac{dG[\tau]}{de_q[\tau]} = \begin{cases} -\frac{1}{\exp(e_q[\tau])} & q = \text{argmax}_j\, e_j[\tau] \\ 0 & \text{otherwise} \end{cases}$$

So we have

$$\frac{d\mathcal{L}}{de_q[\tau]} = \begin{cases} 1 & q = \text{argmax}_j\, e_j[\tau] \text{ but } q \neq q_\tau \\ -1 & q \neq \text{argmax}_j\, e_j[\tau] \text{ but } q = q_\tau \\ 0 & \text{otherwise} \end{cases}$$

Problem 4.3

In a Markov model, the state at time t depends only on the state at time $t-1$. A **semi-Markov model** is a model in which the state at time t depends on a short list of recent states. For example, consider a model in which q_t depends on the most recent **two** frames. Let's suppose the model is fully defined by the following three types of parameters:

- **Initial segment probability:** $\pi_{ij} \equiv p(q_1 = i, q_2 = j|\Lambda)$
- **Transition probability:** $a_{ijk} \equiv p(q_t = k|q_{t-1} = j, q_{t-2} = i, \Lambda)$
- **Observation probability:** $b_k(\vec{x}) \equiv p(\vec{x}_t = \vec{x}|q_t = k, \Lambda)$

Design an algorithm similar to the forward algorithm that is able to compute $p(X|\Lambda)$ with a computational complexity of at most $\mathcal{O}\left\{TN^3\right\}$. Provide a proof that your algorithm has at most $\mathcal{O}\left\{TN^3\right\}$ complexity — this can be an informal proof in the form of a bullet list, as was provided during lecture 12 for the standard forward algorithm.

Solution: Define $\alpha_t(i,j) = p(\vec{x}_1, \ldots, \vec{x}_t, q_{t-1} = i, q_t = j|\Lambda)$. Compute it as

- **Initialize**:

$$\alpha_2(i,j) = \pi_{ij} b_i(\vec{x}_1) b_j(\vec{x}_2), \quad 1 \le i,j \le N$$

- **Iterate:**

$$\alpha_t(j,k) = \sum_{i=1}^{N} \alpha_{t-1}(i,j) a_{ijk} b_k(\vec{x}_t), \quad 1 \le t \le T,\ 1 \le j,k \le N$$

- **Terminate:**

$$p(X|\Lambda) = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_T(i,j)$$

The highest-complexity part of the algorithm is the iteration step, which requires:

- for each of T different time steps t,
- for each of N different values of j,
- for each of N different values of k,
- we must compute a summation with N terms,

hence it has $\mathcal{O}\left\{TN^3\right\}$ complexity.

Problem 4.4

Suppose you have a sequence of $T = 100$ consecutive observations, $X = [x_1, \ldots, x_T]$. Suppose that the observations are discrete, $x_t \in \{1, \ldots, 20\}$. You have it on good information that these data can be modeled by an HMM with $N = 10$ states, whose parameters are

- **Initial state probability:** $\pi_i \equiv p(q_1 = i|\Lambda)$
- **Transition probability:** $a_{ij} \equiv p(q_t = j|q_{t-1} = i, \Lambda)$
- **Observation probability:** $b_j(x) \equiv p(x_t = x|q_t = j, \Lambda)$

In terms of these model parameters, and in terms of the forward probabilities $\alpha_t(i)$ and backward probabilities $\beta_t(i)$ (for any values of i, j, t, x that are useful to you), what is $p(q_{17} = 7, x_{18} = 3|x_1, \ldots, x_{17}, x_{19}, \ldots, x_{100}, \Lambda)$?

Solution: Conditional probability = joint / marginal. The joint probability is

$$p(q_{17} = 7, x_1, \ldots, x_{17}, x_{18} = 3, x_{19}, \ldots, x_{100}) = \sum_{j=1}^{10} \alpha_{17}(7) a_{7j} b_j(3) \beta_{18}(j)$$

The marginal is

$$p(x_1, \ldots, x_{17}, x_{19}, \ldots, x_{100}) = \sum_{i=1}^{10} \sum_{j=1}^{10} \sum_{k=1}^{20} \alpha_{17}(i) a_{ij} b_j(k) \beta_{18}(j)$$

So the conditional is

$$p(q_{17} = 7, x_{18} = 3|x_1, \ldots, x_{17}, x_{19}, \ldots, x_{100}) = \frac{\sum_{j=1}^{10} \alpha_{17}(7) a_{7j} b_j(3) \beta_{18}(j)}{\sum_{i=1}^{10} \sum_{j=1}^{10} \sum_{k=1}^{20} \alpha_{17}(i) a_{ij} b_j(k) \beta_{18}(j)}$$