יובל דלוס 305211880
רון עמיר 208345561

זיהוי שפה טבעית – תרגיל 2

1. (5 pts) Prove that any generative probabilistic classification model that assumes that (1) the sequence of labels forms a Markov Chain, and (2) that the generation of x_i is conditionally independent given y_i of all other x_j and y_j, is an HMM model.

1. עבור משפט ותיוגים $(x_1,..,x_n,y_1,\dots,y_n)$ נתון לנו כי התיוגים יוצרים שרשרת מרקוב, אזי,

$$p(y_1,\dots,y_n)=\prod_{i=1}^{n} q(y_i|y_{i-1},\dots,y_{i-j})$$

$$(*)\ j \coloneqq the\ Markov\ chain\ length$$

בנוסף נתון לנו מ-(2) כי:

$$p(x_i|x_1,..,x_n,y_i,..,y_n)=p(x_i|y_i)$$

ובסה"כ נקבל כי:

$$p(x_1,..,x_n,y_1,\dots,y_n)=\prod_{i=1}^{n+1} q(y_i|y_{i-1},\dots,y_{i-j})\cdot\prod_{i=1}^{n} p(x_i|y_i)$$

וזו בדיוק ההגדרה של HMM מסדר j כנדרש. כאשר ההכפלה הראשונה רצה עד $n+1$ בכדי לכלול גם את STOP.

2. (10 pts) Consider this (toy) biological setup:
A cell can be in one of two states - H, for high GC-content, and L for low GC. On each time step the cell produces one nucleotide, A,C,T or G, and might also change its state. The probability of changing from state H to L is 0.5, and from state L to H is 0.4.
In state H the probabilities for producing nucleotides are 0.2 for A, 0.3 for C, 0.3 for G and 0.2 for T. In L the probabilities are 0.3 for A, 0.2 for C, 0.2 for G and 0.3 for T.
Consider the nucleotide sequence $S = ACCGTGCA$. Use the Viterbi algorithm to find the best state-sequence and calculate the probability of S given this state-sequence. Assume the previous state before S was H.

	H	L
H	0.5	0.5
L	0.4	0.6

	H	L
A	0.2	0.3
C	0.3	0.2
G	0.3	0.2
T	0.2	0.3

יובל דלוס 305211880
רון עמיר 208345561

	A	C	C	G	T
H	0.2 $\boldsymbol{H \to H: 0.3}$	max $H \to H: 0.2 \cdot 0.5 \cdot 0.3$ $L \to H\ 0 \cdot 0.4 \cdot 0.2$ $\boldsymbol{H \to H: 0.03}$	max $H \to H: 0.03 \cdot 0.5 \cdot 0.3$ $L \to H: 0.02 \cdot 0.4 \cdot 0.2$ $\boldsymbol{H \to H: 0.0045}$	max $H \to H: 0.0045 \cdot 0.5 \cdot 0.3$ $L \to H: 0.003 \cdot 0.4 \cdot 0.3$ $H \to H: \frac{27}{40000}$	max $H \to H: \frac{27}{40000} \cdot \frac{1}{2} \cdot \frac{1}{5} = 6.75 \cdot 10^{-5}$ $L \to H: \frac{9}{20000} \cdot \frac{2}{5} \cdot \frac{1}{5} = 3.6 \cdot 10^{-5}$ $\boldsymbol{L \to H: 6.75 \cdot 10^{-5}}$
L	0	max $L \to L: 0 \cdot 0.6 \cdot 0.2$ $H \to L: 0.2 \cdot 0.5 \cdot 0.2$ $H \to L: 0.02$	max $L \to L: 0.02 \cdot 0.6 \cdot 0.2$ $H \to L: 0.03 \cdot 0.5 \cdot 0.2$ $H \to L: 0.003$	max $L \to L: 0.003 \cdot 0.6 \cdot 0.2$ $H \to L: 0.0045 \cdot 0.5 \cdot 0.2$ $\boldsymbol{H \to L: \frac{9}{20000}}$	max $L \to L: \frac{9}{20000} \cdot \frac{3}{5} \cdot \frac{3}{10} = 8.1 \cdot 10^{-5}$ $H \to L: \frac{27}{40000} \cdot \frac{1}{2} \cdot \frac{3}{10} = 1.012 \cdot 10^{-4}$ $H \to L: 10.125 \cdot 10^{-5}$

G	C	A
$H \to H: 6.75 \cdot 10^{-5} \cdot \frac{1}{2} \cdot \frac{3}{10} = 10.125 \cdot 10^{-6}$ $L \to H: 10.125 \cdot 10^{-5} \cdot \frac{2}{5} \cdot \frac{2}{10} = 8.1 \cdot 10^{-6}$ $\boldsymbol{H \to H: 10.125 \cdot 10^{-6}}$	$H \to H: 10.125 \cdot 10^{-6} \cdot \frac{1}{2} \cdot \frac{3}{10} = 1.518 \cdot 10^{-6}$ $L \to H: 12.15 \cdot 10^{-6} \cdot \frac{2}{5} \cdot \frac{2}{10} = 9.72 \cdot 10^{-7}$ $H \to H: 1.518 \cdot 10^{-6}$	$H \to H: 1.518 \cdot 10^{-6} \cdot \frac{1}{2} \cdot \frac{2}{10} = 1.518 \cdot 10^{-7}$ $L \to H: 1.518 \cdot 10^{-6} \cdot \frac{2}{5} \cdot \frac{3}{10} = 1.82 \cdot 10^{-7}$ $H \to H: 1.518 \cdot 10^{-6}$
$L \to L: 10.125 \cdot 10^{-5} \cdot \frac{3}{5} \cdot \frac{1}{5} = 12.15 \cdot 10^{-6}$ $H \to L: 6.75 \cdot 10^{-5} \cdot 10^{-5} \cdot \frac{1}{2} \cdot \frac{3}{10} = 10.125 \cdot 10^{-6}$ $L \to L: 12.15 \cdot 10^{-6}$	$L \to L: 12.15 \cdot 10^{-6} \cdot \frac{3}{5} \cdot \frac{1}{5} = 1.458 \cdot 10^{-6}$ $H \to L: 10.125 \cdot 10^{-6} \cdot \frac{1}{2} \cdot \frac{3}{10} = 1.518 \cdot 10^{-6} = 10.125 \cdot 10^{-6}$ $\boldsymbol{H \to L: 1.518 \cdot 10^{-6}}$	$L \to L: 1.518 \cdot 10^{-6} \cdot \frac{3}{5} \cdot \frac{3}{10} = 2.732 \cdot 10^{-7}$ $H \to L: 1.518 \cdot 10^{-6} \cdot \frac{1}{2} \cdot \frac{2}{10} = 1.518 \cdot 10^{-7} = 10.125 \cdot 10^{-6}$ $\boldsymbol{L \to L: 2.732 \cdot 10^{-7}}$

יובל דלוס 305211880
רון עמיר 208345561

סדר המעברים הינו - $HHHHLHHLL$

3. (10 pts) In class we saw the trigram HMM model and the corresponding Viterbi algorithm. We will now make two main changes. First, we will consider a four-gram tagger, where p takes the form:

$$p(x_1 \cdots x_n, y_1 \cdots y_{n+1}) = \prod_{i=1}^{n+1} q(y_i|y_{i-3}, y_{i-2}, y_{i-1}) \prod_{i=1}^{n} e(x_i|y_i) \tag{1}$$

We assume in this definition that $y_0 = y_{-1} = y_{-2} = *$, where $*$ is the START symbol, $y_{n+1} = STOP$, and $y_i \in \mathcal{K}$ for $i = 1 \cdots n$, where $\mathcal{K}$ is the set of possible tags in the HMM.
Second, we consider a version of the Viterbi algorithm that takes as input **an integer** n (and not a sentence $x_1 \cdots x_n$ as we saw in class) and finds

$$\max_{y_1 \cdots y_{n+1}, x_1 \cdots x_n} p(x_1 \cdots x_n, y_1 \cdots y_{n+1})$$

for a four-gram tagger, as defined in Equation 1. $x_1 \cdots x_n$ may range over the values of some fixed vocabulary $\mathcal{V}$. Complete the following pseudo-code of this version of the Viterbi algorithm for this model . The pseudo-code must be efficient.

Input: An integer n, parameters $q(w|t, u, v)$ and $e(x|s)$.
Definitions: Define $\mathcal{K}$ to be the set of possible tags. Define $\mathcal{K}_{-2} = \mathcal{K}_{-1} = \mathcal{K}_0 = \{*\}$, and $\mathcal{K}_k = \mathcal{K}$ for $k = 1 \cdots n$. Define $\mathcal{V}$ to be the set of possible words.

Initialization – Set $\pi(0,*,*,*) = 1, TagToWord = [\quad]$

$for\ i\ in\ \mathcal{K}$:

$TagToWord[i] = \max_{x \in \mathcal{V}} e(x|\mathcal{K}[i])$

Algorithm:

For k = 1, … , n:

$for\ t \in \mathcal{K}_{k-2}, u \in \mathcal{K}_{k-1}, v \in \mathcal{K}_k$:

$$\pi(k,t,u,v) = \max_{\substack{w \in \mathcal{K}_{k-3} \\ z \in \mathcal{K}_{k-4}}} \begin{pmatrix} \pi(k-2,z,w,t) \times \pi(k-1,w,t,u) \times \\ q(v|w,t,u) \times TagToWord\big(index(v)\big) \end{pmatrix}$$

Return $\max_{t \in \mathcal{K}_{k-2}, u \in \mathcal{K}_{k-1}, v \in \mathcal{K}_k} (\pi(n,t,u,v) \times q(Stop|t,u,v)$

יובל דלוס 305211880
רון עמיר 208345561

4)

	total error	known words error	unknown words error
baseline tagging	0.14811123293132666	0.0704399684933048	0.75043630017452
viterbi tagging	0.12867537127479323	0.05446157308428046	0.7041884816753927
viterbi add-one tagging	0.21678461078441147	0.15190728029706313	0.7198952879581152
viterbi pseudo words tagging	0.11791089404963617	0.05412400135028694	0.6125654450261779
viterbi pseudo add-one tagging	0.2028306588258746	0.14853156295712844	0.6239092495636998

ננתח את ה-confusion matrix :

הבלבול הכי גדול שהמודל שלנו מפיק הוא תיוגים של NN לדברים שהם בעצם NP לפי התיוג הנכון שלהם (259 מקרים).

באותו האופן, דברים שהיו אמורים לקבל NN קיבלו תיוג של NP (154 מקרים).

אלה הם שתי הטעויות הגדולות ביותר של המודל והן מערבות את אותו סט תיוגים, ככל הנראה בשפה שלנו יש הרבה גמישות סביב מילים המקבלות את התיוגים הללו, כלומר שני המקרים קורים באופן דומה במשפט תקין, ולכן המודל לא הצליח להבדיל ביניהם מספיק טוב.
במידה והיה ניתן קורפוס אימון יותר גדול, או עם נתונים אחרים, כנראה שהמודל היה מצליח למצוא טעויות בדיוק רב יותר.

בנוסף, הטעות השלישית בשכיחותה היא תיוג של NN למילים שהן JJ (70 מקרים), ובאופן הפוך יש לנו טעות גם כן של 70 מקרים, כנראה גם אלה דברים שנוטים להשתחלף.

ניסיון נוסף לספר את התחזיות הוא לעבור למודל טרי-גרמי ואף יותר על התיוגים והמילים.