11-751 Speech Recognition and Understanding
Homework 2

Released: Wednesday, Oct 2, 2013
Due: Wednesday, Oct 16, 2013 before class

Problem 1. Dynamic Programming

In continuous speech recognition three different kinds of errors can occur: a spoken word is misrecognized as another word (*substitution*), or a spoken word is not recognized all (*deletion*), or a word that was not spoken is recognized (*insertion*). Based on these three error types, the *word error rate* (WER) is defined as a measure of the performance of the speech recognizer:

$$\text{WER} = \frac{\#\text{substitutions} + \#\text{deletions} + \#\text{insertions}}{\#\text{words to be recognized}} \times 100\%$$

Consider the following example:

```
REF: This great machine can recognize speech
HYP: This       machine can wreck     a      nice beach
           DEL                SUB       SUB    INS  INS
```

The WER is:

$$\text{WER} = \frac{2+1+2}{6} \times 100\% = 83\%$$

Make sure you understand that WER is the minimum error rate you can get by aligning the reference and the hypothesis. Otherwise, you might align the reference and the hypothesis like this:

```
REF: This great   machine can       recognize speech
HYP: This machine can     wreck     a         nice   beach
           SUB     SUB     SUB       SUB       SUB    INS
```

and get a wrong WER of 100%.

Now write a program that reads the reference string and the hypothesis string from two files, and calculates the WER and the corresponding alignment between the two strings. You can use any programming language.

Below is an example which uses command line arguments:

Input: `SOME_OS> WER_program ref_file hyp_file`

where ref_file contains the string “This great machine can recognize speech” and hyp_file contains the string “This machine can wreck a nice beach”.

Output:

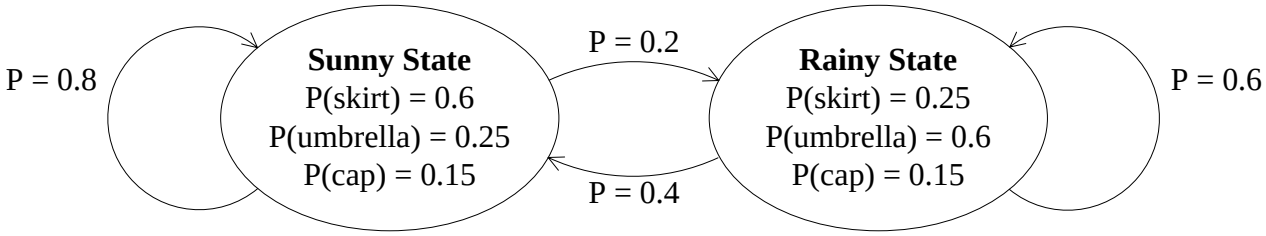
```
REF: This great machine can recognize speech
HYP: This       machine can wreck     a      nice beach
           DEL                SUB       SUB    INS  INS
WER = 83%
```

Please submit your source code and compiled executable (if applicable). You may simplify the output format as long as the alignment is expressed clearly.

Problem 2. Application of Hidden Markov Models

Jessie and you are living on a small island. Assume that the weather is either sunny or rainy. However, you cannot observe the weather directly; you can only see Jessie who visits you every day. What you see is whether she brings an umbrella along, or whether she wears a skirt or a cap. We start on Day 1 in the sunny state and there is one transition per day.

Let Q_t be the state of Day t. For Day 1, we know that $P(Q_1 = \text{sunny}) = 1$ and $P(Q_1 = \text{rainy}) = 0$. These are the initial probabilities. The transition probabilities (what the weather would be like the next day given the weather of the current day) and the emission probabilities (what Jessie would bring or wear given the weather) are shown in the figure below.



Let O_t be Jessie's appearance on Day t. For example, we can infer that $P(O_1 = \text{umbrella}) = 0.25$.

Questions:

1. What is $P(Q_2 = \text{rainy})$?

2. What is $P(O_2 = \text{skirt})$?

3. What is $P(Q_2 = \text{rainy} \mid O_2 = \text{skirt})$?

4. What is $P(O_{100} = \text{cap})$?

5. Let $Y_t = P(Q_t = \text{sunny})$. For example, $Y_1 = 1$. Y_{t+1} can be defined inductively from Y_t by an expression $Y_{t+1} = a + bY_t$. Find the value of a and b.

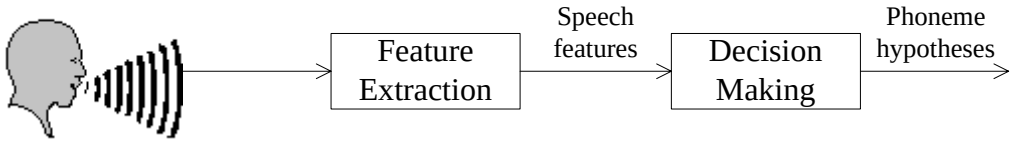
6. Assume that $O_1 = O_2 = O_3 = O_4 = O_5 = \text{umbrella}$. What is the most probable sequence of states? (Hint: This can be solved with the Viterbi algorithm, but it would involve a lot of calculations. Try to answer the question with your intuition, and find a way to justify it.)

Problem 3. The Three Basic Problems of HMMs

Note: Please follow the HMM notations in the Rabiner paper – the classic paper about HMMs. You may use this paper as a guideline for your work on this problem.

In this problem, you will manually solve the three basic problems of HMMs applied on a simplified speech recognition task. You don't need to write any code, but you may want to use Excel for your computation.

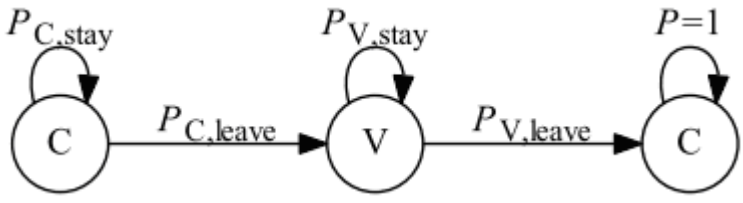
Assume the following speech recognition process:



The front-end component of this speech recognition system is a feature extraction module, which converts speech waveforms into speech feature vectors. The decision making module takes the speech feature vectors as its input and generates phoneme a sequence as the output.

In this homework, let's assume that the system only recognizes two phonemes: C (consonant) and V (vowel). For example, if a person says "bit", the recognizer will output "CVC"; if a person says "agree", the recognizer will output "VCCV". Taking a speech signal as the input, the feature extraction module is assumed to magically output a sequence of numbers in the set {1, 2, 3, 4, 5, 6, 7, 8, 9}. For example, if there are 6 frames of speech, the feature extraction module may generate the sequence 1, 3, 5, 7, 9, 8. The features here are scalars (or 1-dimensional vectors), and they have only nine possible values.

The decision making module makes use of an HMM. The two phonemes, C and V, are each represented by a single state. Each state has two transition probabilities: the probability of staying and the probability of leaving. The states can be concatenated into models of words. For example, the model of the word "CVC" will look like the following. **We require that the first state is the initial state.** The last state will not have a leaving transition, and its staying probability will be one.



The symbols emitted by the states are the feature values (1 to 9). Each of the two states (C, V) has its own emission probability distribution.

Question 1: The Evaluation Problem

Assume the word is "CVC" (whose HMM structure has been shown above), all the transition probabilities are 0.5, and the state emission probabilities are shown in the following table:

Observation symbol *x*	1	2	3	4	5	6	7	8	9
***P*(*x* | C)**	0.20	0.17	0.15	0.13	0.11	0.09	0.07	0.05	0.03
***P*(*x* | V)**	0.03	0.05	0.07	0.09	0.11	0.13	0.15	0.17	0.20

Now we observe the feature sequence 3, 8, 7, 2. Use the forward algorithm to compute the probability of this observation sequence given the HMM. Fill in the following *alpha* trellis to demonstrate your computation:

alpha				
State 3 (C)				
State 2 (V)				
State 1 (C)				
	O_1 = 3	**O_2 = 8**	**O_3 = 7**	**O_4 = 2**

The probability of the observation sequence given the HMM is: _____

Question 2: The Decoding Problem

Given the same HMM and observation sequence as the previous question, use the Viterbi algorithm to find the most probable state sequence given the HMM and observation. Fill in the following trellis to demonstrate your computation. The *delta* in a cell is the probability of the optimal path up to that point, and the *psi* is the back pointer (optimal previous state). You don't need to fill in the *psi* where there is a dash (–).

***delta* (*psi*)**				
State 3 (C)	(–)	(–)	( )	( )
State 2 (V)	(–)	( )	( )	( )
State 1 (C)	(–)	( )	( )	( )
	O_1 = 3	**O_2 = 8**	**O_3 = 7**	**O_4 = 2**

The optimal state sequence is: Q_1 = ____, Q_2 = ____, Q_3 = ____, Q_4 = ____ (choose from 1, 2, 3).

Question 3: The Learning Problem

Given the same HMM observation sequence as the previous question, the forward-backward algorithm maximizes the likelihood of the observation given the model parameters (initial probabilities, transition probabilities, and emission probabilities). Let's manually run an iteration of the forward-backward algorithm to update the transition and emission probabilities. The *alpha* trellis has already been computed in Question 1. In the question, you are asked to fill in the *beta* trellis, compute the *gamma* and *ksi* tables, and compute the updated parameters.

Note: use the definitions of *gamma* and *ksi* in the Rabiner paper, not those in the textbook [Xuedong Huang et al.].

beta				
State 3 (C)				
State 2 (V)				
State 1 (C)				
	$O_1 = 3$	$O_2 = 8$	$O_3 = 7$	$O_4 = 2$

gamma				
State 3 (C)				
State 2 (V)				
State 1 (C)				
	$O_1 = 3$	$O_2 = 8$	$O_3 = 7$	$O_4 = 2$

ksi				
State 3 (C) → 3 (C)				
State 2 (V) → 3 (C)				
State 2 (V) → 2 (V)				
State 1 (C) → 2 (V)				
State 1 (C) → 1 (C)				
	$O_1 = 3$	$O_2 = 8$	$O_3 = 7$	$O_4 = 2$

Since both State 1 and State 3 are the state C, the parameters of the state C should be computed by combining the *gamma* and *ksi* values of both State 1 and State 3. Write down the updated parameters:

Transition probabilities:

$P_{C,stay}$ = _____ $P_{C,leave}$ = _____

$P_{V,stay}$ = _____ $P_{V,leave}$ = _____

Emission probabilities:

Observation symbol *x*	1	2	3	4	5	6	7	8	9
***P*(*x* | C)**									
***P*(*x* | V)**									

Question 4: Bayesian Decision

The Bayesian decision rule for our C-V speech recognizer is as follows:

$$\text{Best phoneme sequence} = \arg\max_i P(\text{phoneme sequence}_i \mid \text{observation})$$
$$= \arg\max_i P(\text{observation} \mid \text{phoneme sequence}_i) \cdot P(\text{phoneme sequence}_i)$$

The language model, $P(\text{phoneme sequence}_i)$, is assumed to be a constant for every phoneme sequence, so this term can be ignored in the decision rule. The acoustic model probability, $P(\text{observation} \mid \text{phoneme sequence}_i)$, is given by the HMM of the phoneme sequence.

Suppose that besides the phoneme sequence CVC we've been working on, the acoustic model also contains a phoneme sequence consisting of a single C. The HMM of the second phoneme sequence will have only one state C, with a self-loop probability of 1. Given the observation sequence 3, 8, 7, 2, which of the two phoneme sequences will be recognized by the Bayesian decision rule? Why? (Use the original model parameters for the calculations, not the updated parameters you calculated in Question 3)