**11-751 Speech Recognition and Understanding**

Nikolas Wolfe

Homework 2

October 16th, 2013

**Problem 1: Dynamic Programming**

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.

Answer: The code will be submitted separately with this homework, but pasting the essential components here…

This is written in JavaScript using JQuery. The HTML markup contains two text input elements containing the reference and hypothesis strings to be evaluated, and a button (with id=‘calc’) to initiate the alignment problem, called by the following function:

$('#calc').click( …code here… )

Then the distance calculation is done by a call to function dist\_matrix(ref, hyp) { … }

<script>

$(document).ready(function() {

// some constants

var MATCH = "MATCH";

var DEL = "DEL";

var INS = "INS";

var SUB = "SUB";

var dtw, ref, hyp;

// hide distance matrix button

$('#matrix').hide();

// When the word error rate button is clicked

$('#calc').click( function() {

// trim all whitespace and split by word boundaries

ref = $.trim($('#ref').val().replace(/\s+/g, " ")).split(" ");

hyp = $.trim($('#hyp').val().replace(/\s+/g, " ")).split(" ");

dtw = dist\_matrix(ref, hyp);

var d = dtw[dtw.length - 1][dtw[0].length - 1];

var wer = ((d[0] / ref.length) \* 100).toFixed(0);

$('#wer').html( "WER: " + wer + "%");

$('#align').html( generate\_alignment( dtw, ref, hyp ) );

$('#matrix').show();

});

… code here …

// calculate the distance matrix

function dist\_matrix(ref, hyp) {

// initialize the distance array

var dtw = new Array( ref.length + 1 );

$.each(dtw, function( i ) {

dtw[i] = new Array( hyp.length + 1 );

$.each(dtw[i], function( j ) {

dtw[i][j] = [0,0,0,""];

});

});

// initialize edge weights in the array

var i, j;

$.each(dtw, function( i ) {

dtw[i][0][0] = i;

});

$.each(dtw[0], function( j ) {

dtw[0][j][0] = j;

});

/\* calculate the distance matrix

\*

\* NOTE: this draws inspiration from the Wikipedia article on

\* Levenshtein Distance. here: http://en.wikipedia.org/wiki/Levenshtein\_distance

\*/

var m, n;

m = dtw.length;

n = dtw[0].length;

for( i = 1; i < m; i++ ) {

for( j = 1; j < n; j++ ) {

var r, h;

r = ref[i - 1].toLowerCase();

h = hyp[j - 1].toLowerCase();

if( r == h ) { // match

// take diagonal distance, save back pointer

dtw[i][j] = [dtw[i - 1][j - 1][0],(i - 1),(j - 1),"MATCH"];

} else { // mismatch

var del, ins, sub, min;

del = dtw[i - 1][j][0] + 1; // deletion

ins = dtw[i][j - 1][0] + 1; // insertion

sub = dtw[i - 1][j - 1][0] + 1; // substitution

// find the minimum distance, save the back-pointers

min = Math.min( del, ins, sub );

if( min == del ) {

dtw[i][j] = [min,(i - 1),(j),DEL];

}

else if( min == ins ) {

dtw[i][j] = [min,(i),(j - 1),INS];

}

else {

dtw[i][j] = [min,(i - 1),(j - 1),SUB];

}

}

}

}

return dtw;

}

… code here …

</script>

**Problem 2: Application of Hidden Markov Models**

For Day 1, we know that P(Q1 = Sunny) = 1 and P(Q1 = Rainy) = 0.



Let Ot be Jessie’s appearance on Day t. For example, we can infer that P(O1 = umbrella) = 0.25.

Questions:

1. What is P(Q2 = Rainy)?

Because Q1 = Sunny, we know that the transition probability of switching to Rainy state is P=0.2, so the likelihood that Q2 is Rainy is P(Q2=Rainy |Q1=Sunny), which is 0.2.

1. What is P(O2 = Skirt)?

In both Rainy and Sunny state we have probabilities for skirts, so assuming we are in Sunny state in Q1, we have the transition probability of remaining in Sunny state, which is P=0.8, and the probability of observing ‘skirt’ given a Sunny state is 0.6. This means we have P(Q2 = Sunny *and* O2 = Skirt), which is P(Q2=Sunny) x P(O2=Skirt|Q2=Sunny), which is 0.8 \* 0.6, which is 0.48. If we switch to Rainy state, we have a transition probability of 0.2, and a skirt observation probability of 0.25, which is P(Q2=Rainy) x P(O2=Skirt|O2=Rainy), which is 0.2 \* 0.25, which is 0.05. So, the overall likelihood of P(O2=Skirt) is P(Q2 = Sunny *and* O2 = Skirt) *or* P(Q2 = Rainy *and* O2 = Skirt), which is 0.48 + 0.05, which is 0.53.

1. What is P(Q2 = Rainy | O2 = Skirt)?

From Question 1, the probability of P(Q2 = Rainy) = 0.2. The probability of skirt is based on the overall probability of observing skirt in O2, which we determined in Question 2 = 0.53. So, P(Q2 = Rainy | O2 = Skirt) is P(O2=Skirt *and* Q1=Rainy)/P(Q2=Skirt), which is (0.53 \* 0.2)/0.53 = 0.2.

1. What is P(O100 = cap)?

All trials are dependent only on the previous trial, so the 100th state probability of ‘cap’ is equivalent to the overall likelihood of cap, which is the probability of observing cap in O100 given that Q99 was Sunny, *or* the probability of observing cap in O100 given that Q99 was Rainy. If Q99 was Rainy, then we either transitioned into Sunny in Q100 or we remained Rainy in Q100. Likewise if Q99 was Sunny, then we either transitioned into Rainy in Q100 or we remained in Sunny in Q100.

First assume Q99 was Sunny, i.e. P(Q99 = Sunny) = 1.

1. P(Q100 = Sunny *and* O100 = Cap) = P(Q100=Sunny) x P(O100=Cap|Q100=Sunny) = 0.8 \* 0.15 = 0.12
2. P(Q100 = Rainy *and* O100 = Cap) = P(Q100 = Rainy) x P(O100 = Cap|Q100 = Rainy) = 0.2 \* 0.15 = 0.03
3. P(O100 = Cap | Q99 = Sunny) = a.) *or* b.) = 0.12 + 0.03 = 0.15

Second, assume Q99 = Rainy, i.e. P(Q99 = Rainy) = 1.

1. P(Q100 = Sunny *and* O100 = Cap) = P(Q100 = Sunny) x P(O100 = Cap | Q100 = Sunny) = 0.4 \* 0.15 = 0.06
2. P(Q100 = Rainy *and* O100 = Cap) = P(Q100 = Rainy) x P(O100 = Cap | Q100 = Rainy) = 0.6 \* 0.15 = 0.09
3. P(O100 = Cap | Q99 = Sunny) = d.) *or* e.) = 0.06 + 0.09 = 0.15

Overall, P(O100 = Cap) = c.) *or* f.) = 0.15 + 0.15 = 0.30

And, this makes intuitive sense, because regardless of what state we are in, the probability of Cap is always 0.15, so P(Cap|Sunny) or P(Cap|Rainy) = 0.15 + 0.15 = 0.30

1. Let Yt = P(Qt = Sunny). For example, Y1 = 1. Yt+1 can be defined inductively from Yt by an expression Yt+1 = a + BYt. Find the value of *a* and *b*.

Yt = P(Qt = Sunny), so Y1 = 1 => P(Q1 = Sunny) = 1,

Yt+1 = a + BYt => a + B(P(Qt = Sunny))

By induction,

Yt+1 = a + BYt

Yt+2 = a + BYt+1

Yt+3 = a + BYt+2

…

Yt+n = a + BYt+n-1

And…

Y1 = 1

Y2 = a + B(1)

Y3 = a + B(a + B)

Y4 = a + B(a + B(a + B))

Y5 = a + B(a + B(a + B(a + B)))

…

Yt+1 = a + B(Yt )

We know that Y1 = 1 = Sunny, which means the probability of remaining in that state in Y2 is 0.8.

So Y2 = a + B(1) = P(Q2 = Sunny) = 0.8

Then Y3 = a + B(Y2), where Y2 = a + B = 0.8

Which is Y3 = a + B(0.8)

But Y3 is the probability of remaining in Sunny (0.8)given that Y2 is Sunny OR the probability that in Y2 the state was Rainy (0.2 from Y1) and then switched *back* to Sunny in Y3, a probability of 0.4. So the probability of P(Q3 = Sunny |Q2 = Rainy) = 0.4, which of course has an 0.2 chance of happening *given* that Q1 = Sunny. So in Y3 the probability of getting Sunny is the likelihood of the sequence:

Q1 Q2 Q3 = SSS, which is (1 \* 0.8 \* 0.8 = 0.64)

Or the sequence Q1 Q2 Q3 = SRS, which is (1 \* 0.2 \* 0.4 = 0.08),

So Y3 = P(SSS) + P(SRS) = 0.64 + 0.08 = 0.72

Y2 = a + B = 0.8

Y3 = a + B(0.8) = 0.72

Two equations, two unknowns...

a = 0.4

B = 0.4

*Proof by Induction:*

|  |  |
| --- | --- |
| **Yt+1 = 0.4 + 0.4Yt** | State Transition Probability Chains… |
| Y1 = **1** | Q1 = S = **1** |
| Y2 = 0.4 + 0.4(1) = **0.8** | Q1Q2 = SS = 1 \* 0.8 = **0.8** |
| Y3 = 0.4 + 0.4(0.8) = **0.72** | Q1Q2Q3 = SSS = 1 \* 0.8 \* 0.8 = 0.64  Q1Q2Q3 = SRS = 1 \* 0.2 \* 0.4 = 0.08  0.08 + 0.64 = **0.72** |
| Y4 = 0.4 + 0.4(0.72) = **0.688** | Q1Q2Q3Q4 = SSSS = 1 \* 0.8 \* 0.8 \* 0.8 = 0.512  Q1Q2Q3Q4 = SRSS = 1 \* 0.2 \* 0.4 \* 0.8 = 0.064  Q1Q2Q3Q4 = SSRS = 1 \* 0.8 \* 0.2 \* 0.4 = 0.064  Q1Q2Q3Q4 = SRRS = 1 \* 0.2 \* 0.6 \* 0.4 = 0.048  0.512 + 0.064 + 0.064 + 0.048 = **0.688** |

Etc. and so on… So,

**Yt+1 = 0.4 + 0.4Yt**

1. Assume that O1 = O2 = O3 = O4 = O5 = 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.)

Intuition tells me that the most likely state to emit an umbrella in Q1 is Rainy state, with a probability of 0.6, compared with Sunny state, which has a probability of 0.25. Also, the likelihood of remaining in Rainy state is 0.6 combined with an emission probability of 0.6 for umbrella, so the likelihood of Q2 emitting an umbrella as well is 0.6\*0.6 = 0.36. If the system were to transition to Sunny, probability would be 0.4\*0.25 = 0.1. Furthermore, even if we were in Sunny state to begin with, the probability of remaining in Sunny state and seeing an umbrella is 0.8 \* 0.25 = 0.2. Any transition to Sunny state involving the viewing of umbrella is going to have a smaller probability of seeing umbrella again than remaining continuously in Rainy state, ergo, intuition tells me that the highest probability of viewing O1 = O2 = O3 = O4 = O5 = umbrella is for Q1 = Q2 = Q3 = Q4 = Q5 = Rainy, that is, a succession of Rainy states.

**Problem 3: The Three Basic Problems of HMMs**

Given the following HMM:



**Question 1: The Evaluation Problem**

Given the following alpha trellis for Observation {3, 8, 7, 2},

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STATE 3 ( C ) | 0.15 | 0.05 | 0.07 | 0.17 |
| STATE 2 ( V ) | 0.07 | 0.17 | 0.15 | 0.05 |
| STATE 1 ( C ) | 0.15 | 0.05 | 0.07 | 0.17 |
|  | 3 | 8 | 7 | 2 |

The probability of state (Qt) transition sequences, where Q1 = State 1 (C), Q2 = State 2 (V), and Q3 = State 3 (C), is as follows, (Note: the sequence cannot go backwards, it can only remain in its current state or go forward, and all transitions are equally likely (0.5)):

|  |  |
| --- | --- |
| Q Sequence | Probability |
| Q1 Q1 Q1 Q1 | 1.11563E-05 |
| Q1 Q1 Q1 Q2 | 3.28125E-06 |
| Q1 Q1 Q2 Q2 | 7.03125E-06 |
| Q1 Q2 Q2 Q2 | 2.39063E-05 |
| Q1 Q2 Q2 Q3 | 8.12813E-05 |
| Q1 Q2 Q3 Q3 | 3.79313E-05 |
| Sum: | 0.000164588 |

The probability of the observation sequence is: **0.000164588**.

**Question 2: The Decoding Problem**

Using the same HMM observation sequence as the previous question, use the Viterbi algorithm to find the most probably state sequence given the HMM and observation.

The delta (psi) trellis is as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STATE 3 ( C ) | 0.15 (-) | 0.05 (-) | 0.00044625 (2) | **8.12813E-05 (2)** |
| STATE 2 ( V ) | 0.07 (-) | **0.01275 (1)** | **0.00095625 (2)** | 2.39063E-05 (1) |
| STATE 1 ( C ) | **0.15 (-)** | 0.00375 (1) | 0.00013125 (1) | 1.11563E-05 (1) |
|  | 3 | 8 | 7 | 2 |
|  | MAX: | 0.01275 | 0.00095625 | 8.12813E-05 |

Looking at the maximum delta values in each column the table, we see that there is an optimal state sequence, which is:

*Q1* **= (1),***Q2* **= (2),***Q3* **= (2),***Q4* **= (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 give the model parameters. Run a manual iteration of the forward-backward algorithm to update the transition and emission probabilities.

The beta trellis:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STATE 3 ( C ) | 0.000074375 | 0.002975 | 0.085 | 1 |
| STATE 2 ( V ) | 0.0010285 | 0.011225 | 0.11 | 1 |
| STATE 1 ( C ) | 0.001256625 | 0.0121 | 0.11 | 1 |
|  | 3 | 8 | 7 | 2 |

The gamma trellis:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STATE 3 ( C ) | 0.041069226 | 0.055879038 | 0.1973466 | 0.435897436 |
| STATE 2 ( V ) | 0.265033408 | 0.716848234 | 0.547263682 | 0.128205128 |
| STATE 1 ( C ) | 0.693897366 | 0.227272727 | 0.255389718 | 0.435897436 |
| OBSERVATION | 3 | 8 | 7 | 2 |
| *SUM:* | *1* | *1* | *1* | *1* |

The ksi trellis:

|  |  |  |  |
| --- | --- | --- | --- |
| STATE 3 > 3 | 0.041 | 0.056 | 0.197 |
| STATE 2 > 3 | 0.019 | 0.190 | 0.423 |
| STATE 2 > 2 | 0.246 | 0.527 | 0.124 |
| STATE 1 > 2 | 0.527 | 0.155 | 0.058 |
| STATE 1 > 1 | 0.167 | 0.072 | 0.197 |
| OBSERVATION | 3 | 8 | 7 |
| *SUM* | *1* | *1* | *1* |

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:

|  |  |
| --- | --- |
| PC,stay = | 0.49698545246 |
| PV,stay = | 0.586671109 |
| PC,leave = | 0.628834414 |
| PV,leave = | 0.413328891 |

Adjusted Emission probabilities:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Observation | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | SUM |
| P(x|C) | 0 | 0.37 | 0.31 | 0 | 0 | 0 | 0.19 | 0.12 | 0 | 1 |
| P(x|V) | 0 | 0.08 | 0.16 | 0 | 0 | 0 | 0.33 | 0.43 | 0 | 1 |

**Question 4: Bayesian Decision**

Best phoneme sequence = argmaxi P(phoneme sequence | observation )

= argmaxi P(observation | phoneme sequencei ) \* P(phoneme sequencei )

The language model P(phoneme sequencei ), is assumed to be a constant for every phoneme sequence, so this term can be ignored. Assume the acoustic model 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?

*Answer:*

HMM 1 = 3 State, CVC = **0.000164588**

HMM 2 = 1 State, C = 0.15 \* 0.05 \* 0.07 \* 0.17 = **0.00008925**

The probability of phoneme sequence **CVC** is greater than the probability of phoneme sequence **C**, so the Bayesian decision rule will accept the **CVC** sequence.