PART IIB 2010 4F11: SPEECH AND LANGUAGE PROCESSING DR W J BYRNE

1. This question concerns the use of linear predictive coding (LPC) in modelling speech. The order of the LPC model is p. The n^{th} sample of the speech signal is s_n.

(a) Write an expression for the prediction error e_n in terms of the speech signal s_n and the linear prediction coefficients $a_1,\dots,a_p$. [15%]

(from slide 11 of Lectures 2/3)

$$\begin{aligned}\hat{s}_n &= a_1 s_{n-1} + a_2 s_{n-2} + \ldots + a_p s_{n-p} \\ \hat{s}_n &= \sum_{i=1}^{p} a_i s_{n-i} \\ e_n &= s_n - \hat{s}_n \\ &= s_n - \sum_{i=1}^{p} a_i s_{n-i}\end{aligned}$$

(b) Show how the transfer function $A(z)$ of the prediction error filter relates the speech spectrum $S(z)$ to the transform of the prediction error $E(z)$. [15%]

(from slide 12 of Lectures 2/3)

Taking z-transforms of both sizes we obtain

$$\begin{aligned}E(z) &= \left[1 - \sum_{i=1}^{p} a_i z^{-i}\right] S(z) \\ \frac{S(z)}{E(z)} &= \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}}\end{aligned}$$

(c) Derive the *normal equations*

$$\sum_n s_n s_{n-j} = \sum_{k=1}^{p} a_k \sum_n s_{n-k} s_{n-j} \quad j = 1,\dots,p$$

which are solved to obtain the optimum filter coefficients under the mean square error criterion. [20%]

(from slide 15 of Lectures 2/3)

The summed squared prediction error E_T

$$\begin{aligned}E_T &= \sum_n e_n^2 \\ &= \sum_n (s_n - \sum_{k=1}^{p} a_k s_{n-k})^2\end{aligned}$$

To minimise E_T find the solution to $\partial E_T / \partial a_k = 0$.

$$\begin{aligned} \frac{\partial E}{\partial a_j} = 0 &= -\sum_n \left(2(s_n - \sum_{k=1}^{p} a_k s_{n-k}) s_{n-j} \right) \\ &= -2 \sum_n s_n s_{n-j} + 2 \sum_n \sum_{k=1}^{p} a_k s_{n-k} s_{n-j} \end{aligned}$$

Rearranging gives the set of p simultaneous linear equations (normal equations)for values of j from 1 to p

(d) Explain the assumptions underlying the *autocorrelation method* and give the simplified form of the normal equations which result. [20%]

(from slide 18, Lectures 2/3)

A window is applied to the speech so that

$$s_n = 0 \text{ if } n < 0 \text{ or } n >= N$$

Defining the autocorrelation

$$r_k = \sum_{n=0}^{N-1-k} s_n s_{n+k}$$

Now the normal equations are as shown below (a Toeplitz matrix):

$$\begin{pmatrix} r_1 \\ r_2 \\ \cdots \\ r_p \end{pmatrix} = \begin{pmatrix} r_0 & r_1 & r_2 & \cdots & r_{p-1} \\ r_1 & r_0 & r_1 & \cdots & r_{p-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ r_{p-1} & r_{p-2} & \cdots & \cdots & r_0 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \cdots \\ a_p \end{pmatrix}$$

(e) Give Durbin's Algorithm for finding the optimum filter coefficients and explain how they guarantee that increasing the order of the LPC predictor reduces the mean square error. [30%]

(from slide 20, Lectures 2/3)

Denoting the values of the LP parameters at iteration i by $a_k^{(i)}$ and the sum-squared predictor error (or residual energy) by $E_T^{(i)}$ ($E_T^{(0)} = r_0$) for i = 1, 2, ...

$$\begin{aligned} k_i &= \left(r_i - \sum_{j=1}^{i-1} a_j^{(i-1)} r_{i-j} \right) / E_T^{(i-1)} \\ a_i^{(i)} &= k_i \\ a_j^{(i)} &= a_j^{(i-1)} - k_i a_{i-j}^{(i-1)} \qquad 1 \leq j < i \\ E_T^{(i)} &= (1 - k_i^2) E_T^{(i-1)} \end{aligned}$$

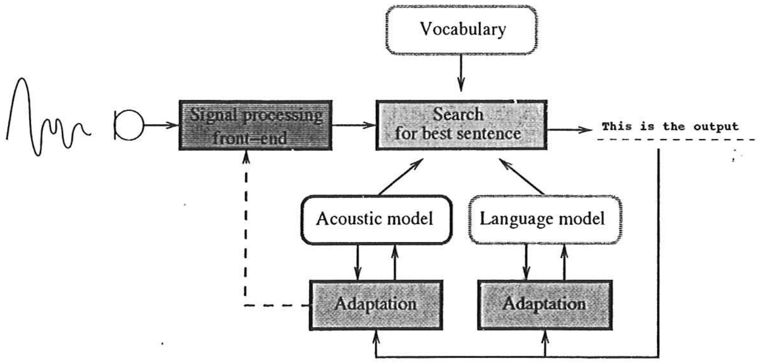
$E_T^{(i)}$ is the MSE for the order i predictor. Since $k_i \leq 1$ the MSE prediction error decreases unless $k_i = 1$.

Assessor's comment:

Not an easy question although one student received full marks. Most answers displayed sound knowledge of discrete systems but many were weaker in its use for linear prediction.

2 (a) Draw a block diagram of the generic speech recognition architecture and give a brief description of each component. [20%]

(from slide 7, Lectures 4/5)



(b) A speech recognition system is to be constructed using Hidden Markov Models (HMMs). The HMMs will have Gaussian observation distributions and will be trained as whole-word models so that each word has its own HMM. A training set of R utterances for each word in the recognition vocabulary is available, and the Baum Welch algorithm is to be used to estimate the parameters of the HMM observation distributions.

(i) Suggest an initialisation procedure for the parameters of the Gaussian observation distributions. [10%]

A sample mean and covariance can be computed over all available data for each word and these can be used to initialize the distributions.

(ii) Give the equations defining the forward and backward probabilities used in the Baum Welch algorithm. Derive the recursions for one of the probabilities. [20%]

(from slides 27-31, Lectures 4/5)

$\alpha_j(t)$ is the likelihood of the HMM producing all observations up to time t and occupying state j at time t; it is called the forward probability:

$$\alpha_j(t) = p(\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_t, x(t) = j \,|\, \lambda)$$

The Backward probability is defined as

$$\beta_j(t) = p(\mathbf{o}_{t+1}, \mathbf{o}_{t+2}, \ldots, \mathbf{o}_T \,|x(t) = j, \lambda)$$

They satisfy the following recursions:

$$\alpha_j(t) = b_j(\mathbf{o}_t) \left[\sum_{k=1}^{N-1} \alpha_k(t-1) a_{kj} \right]$$

$$\beta_j(t) = \sum_{k=2}^{N-1} a_{jk} b_k(\mathbf{o}_{t+1}) \beta_k(t+1)$$

(iii) Give a relationship for the probability of being in state j at time $t, L_j(t)$, in terms of the forward and backward probabilities. [10%]

Since

$$p(x(t) = j, \mathbf{O} \,|\, \lambda) = \alpha_j(t) \beta_j(t)$$

it follows that

$$L_j(t) = P(x(t) = j \,|\, \mathbf{O}, \lambda) = \frac{1}{p(\mathbf{O}|\lambda)} \alpha_j(t) \beta_j(t)$$

(iv) Give the Baum Welch reestimation formulae for the mean and variance parameters of the Gaussian observation distribution associated with each state. [20%]

(from slide 32, Lectures 4/5)

$$\hat{\mu}_j = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T^{(r)}} L_j^{(r)}(t) \mathbf{o}_t^{(r)}}{\sum_{r=1}^{R} \sum_{t=1}^{T^{(r)}} L_j^{(r)}(t)}$$

$$\hat{\Sigma}_j = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T^{(r)}} L_j^{(r)}(t) (\mathbf{o}_t^{(r)} - \hat{\mu}_j)(\mathbf{o}_t^{(r)} - \hat{\mu}_j)'}{\sum_{r=1}^{R} \sum_{t=1}^{T^{(r)}} L_j^{(r)}(t)}$$

(c) A large-vocabulary isolated-word system is to be trained for use with the Google Maps application. It is decided to use HMMs based on word-internal triphones.

(i) Discuss why word-internal triphones might be a better choice for this application than whole-word HMMs. [10%]

Triphones are shared across words so that a model need not be trained for every word in the vocabulary. This reduces run-time cost in the calculation of likelihood and also improves robustness in that triphones can be clustered so that all are adequately trained whereas many words in the vocabulary will have few or no training samples.

(ii) Contrast the HMM training procedure for word-internal triphone models to the training procedure for whole-word acoustic models. [10%]

(from slide 17, Lecture 6)

The usual steps involved in training triphone models are to train monophone models using BW; clone these to triphone models; continue BW training; cluster the triphone states, using decision tree or other methods; continue BW training of the clustered triphone models. Additional state-level Gaussian mixtures are added, although this can also be done with whole-word models.

Assessor's comment:

Most students apparently found this question easy, although many found part c difficult. The question asked for a comparison of 'whole word' vs. triphone HMMs. Most students clearly discussed the difference between these models, but many were unable to cast these differences in terms of strengths and weaknesses for acoustic modeling.

3. (a) What is a weighted finite state acceptor? What is the main difference between a weighted finite state acceptor (WFSA) and a weighted finite state transducer (WFST)? [20%]

A weighted acceptor over a finite input alphabet Σ is a finite directed graph with a set of nodes Q (states) and a set of arcs E (edges).

- Each arc (or edge) e has an initial (or start) state $s(e)$ and a final state $f(e)$.
- Each arc e is labeled with an input symbol $i(e)$ and a weight $w(e)$.
- The weights take values in $\mathbb{K}$

The main difference between an Acceptor and a Transducer is that each arc e of a Transducer has also an output symbol $o(e) \in \Delta$, where Δ is the output alphabet.

(Lecture 9-10, slides 7 and 23)

(b) Weighted finite state acceptors assign a weight to a particular string by summation ($\oplus$ operation) over the weights of all paths that generate the string, where the weight of each path is obtained as the product ($\otimes$ operation) of the path arc weights and the path initial and final weights.

(i) Define a semiring in the context of WFSAs. [10%]

A *Semiring* is defined by a sum $\oplus$ operation, a product $\otimes$ operation, and two identity elements $\bar{0}$ and $\bar{1}$, such that:

- For a weight $k \in \mathbb{K}$: $\bar{0} \oplus k = k$; $\bar{1} \otimes k = k$; $\bar{0} \otimes k = \bar{0}$
- $\oplus$ and $\otimes$ distribute and commute in the familiar way **(Lecture 9-10, slide 9)**

(ii) Complete Table 1 to describe the attributes of three commonly used semirings. [20%]

Semiring	$\mathbb{K}$	$\oplus$	$\otimes$	$\bar{0}$	$\bar{1}$
Probability					
Log					
Tropical					

Table 1

Semiring	$\mathbb{K}$	$\oplus$	$\otimes$	$\bar{0}$	$\bar{1}$
Probability	$\mathbb{R}_+$	$+$	$\times$	0	1
Log	$\mathbb{R} \cup \{-\infty, \infty\}$	$\oplus_{\log}$	$+$	∞	0
Tropical	$\mathbb{R} \cup \{-\infty, \infty\}$	min	$+$	∞	0

$\oplus_{\log} : k_1 \oplus_{\log} k_2 = -\log(e^{-k_1} + e^{-k_2})$

(Lecture 9-10, slide 9)

(iii) A WFSA A can generate the sequence 'a b' via two alternative paths. What does the weight, $[\![A]\!]$('a b'), assigned to the sequence 'a b' by A represent in each of the three semirings of Table 1? [30%]

- In the Probability Semiring, $[\![A]\!]$('a b') is the marginal probability of the sequence given the two alternative paths: $p_1(\text{'a b'})\,p_1 + p_2(\text{'a b'})\,p_2$
- In the Log Semiring, it is the negative log of the marginal probability from above
- In the Tropical Semiring, it is the negative log Viterbi likelihood: $-\max\left[\log p_1(\text{'a b'})\,p_1\,,\,\log p_2(\text{'a b'})\,p_2\right]$

(Lecture 9-10, slides 11-13)

(c) Two WFSTs A and B are shown in Fig. 1, where A maps x to y and B maps y to z. Draw the transducer $A \circ B$, which maps x to z, that results from the composition of A with B in the tropical semiring. [20%]

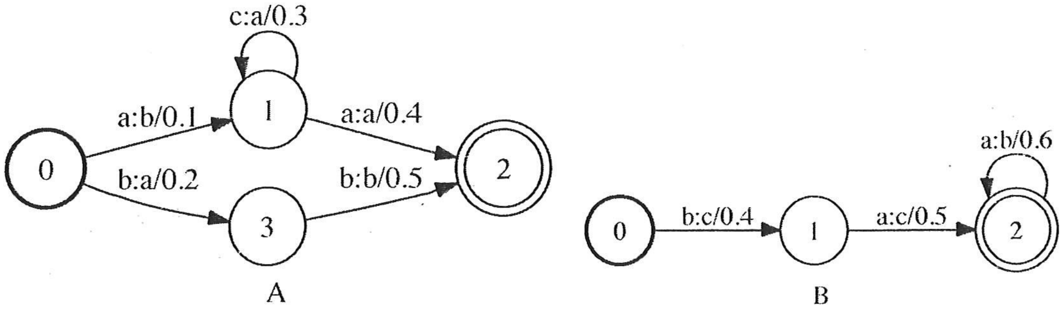
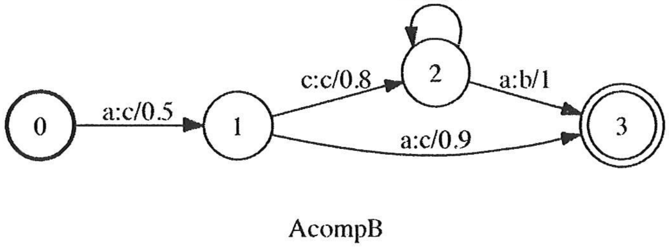


Fig. 1

The $A \circ B$ is:



(Lecture 9-10, slide 27)

Assessor's comment:

Section (c) on drawing the result of a composition of transducers was very difficult - two students did well but the rest struggled.

4 This question concerns the use of alignment models in statistical machine translation. An English sentence of I words is denoted by the sequence $e_0^I = e_0, e_1, \ldots, e_I$, where e_0 is the additional NULL word added to the start of the sentence. A foreign sentence of J words is denoted $f_1^J = f_1, \ldots, f_J$. Alignment between the two sentences is specified by the sequence $a_1^J = a_1, \ldots, a_J$.

(a) By making simplifying conditional independence assumptions, describe the translation probability distribution $P(f_1^J, a_1^J, J | e_0^I)$ in terms of its three component distributions: the sentence length distribution, the word translation distribution, and the word alignment distribution. [20%]

Making simplifying conditional independence assumptions:

$$\begin{aligned} P(f_1^J, a_1^J, J | e_0^I) &= P(f_1^J | J, a_1^J, e_0^I) \quad P(a_1^J | J, e_0^I) \quad P(J|I) \\ &= \textstyle\prod_{j=1}^J P_T(f_j | e_{a_j}) \quad P_A(a_1^J | J, I) \quad P_L(J|I) \end{aligned}$$

where:

- $P_L(J|I)$ is the Sentence Length Distribution
- $P_T(f|e)$ is the Word Translation Distribution
- $P_A(a_1^J | J, I)$ is the Word Alignment Distribution

(Lecture 12 (SMT-Alignment), slide 2)

(b) Give the formulae of the alignment distribution under IBM models 1 and 2 and the HMM alignment model. Explain their differences. [20%]

Formulae:

- Model-1:

$$P_A(a_1^J | J, I) = \frac{1}{I^J}$$

- Model-2:

$$P_A(a_1^J | J, I) = \prod_{j=1}^J p_{M2}(a_j | j, J, I)$$

- HMM Model:

$$P_A(a_1^J | J, I) = \prod_{j=1}^J p_{HMM}(a_j | a_{j-1}, I)$$

Differences:

- Model-1 assumes that the link distribution is entirely flat (all positions equally probable)

- Model-2 assumes that the link distribution depends on the foreign word location j
- HMM-Model assumes that the link distribution depends on the link of the previous word (that is, it has a history of one link)

(Lecture 12 (SMT-Alignment), slides 4-5)

(c) Alignment link sets B and B' are extracted from two different alignments. How is the alignment error between B and B' computed? Why is the alignment error useful for developing statistical machine translation systems? [20%]

The Alignment Error AE(B,B') is computed as follows:

$$AE(B,B') = \frac{|B'|+|B|-2|B\cap B'|}{|B'|+|B|}$$

ignoring NULL word links in B anb B'.

AE can be an indicator of the quality of word alignment models, i.e., large reductions in AE are often correlated with improvements in translation quality. AE is often used as an intermediate quality measure in translation system development.

(Lecture 13 (SMT-Translation), slide 10)

(d) How are 'phrases' defined in the context of phrase-based statistical machine translation models? Describe the advantages of using phrases rather than words in a statistical machine translation system. [20%]

- In this context, 'phrases' are simply word sequences extracted from sentences. A phrase is a sequence of words which can be translated.
- Phrases can capture local syntactic and semantic context that words cannot capture.

(Lecture 13 (SMT-Translation), slide 11)

(e) Briefly describe how a word-based alignment model can be used to extract phrase pairs from a parallel corpus. [20%]

- Phrase pairs are extracted from word-level alignments, typically generated using IBM-4 on the same parallel text over which it was trained
- Phrase Pairs cover patterns of word alignments in the training bitext
- Rules are defined to specify valid phrase pairs, for example:

Two phrases are aligned if their words align only with each other

(Lecture 13 (SMT-Translation), slide 12)

Assessor's comment:

Quite an easy question, for those who tried it.