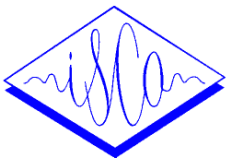

2nd International Conference on
Spoken Language Processing (ICSLP 92)
Banff, Alberta, Canada
October 12-16, 1992

A HIDDEN MARKOV MODEL STRUCTURE FOR THE ACQUISITION OF SPEECH BY MACHINE, ASM

Frank Fallside

Cambridge University Engineering Department
Trumpington Street, Cambridge CB2 1PZ, UK

ABSTRACT

A structure for the acquisition of speech by machines, asm, employing neural networks, for the 'simultaneous' training of recognition and synthesis has been presented [1]. In the present paper it is shown that there is an equivalent structure employing HMMs. By analogy with recent results relating HMM and neural network architectures it is seen that the two methods for asm are equivalent.

I. INTRODUCTION

It is apparent that humans acquire speech by learning to classify the speech sounds of their language from existing speakers and at the same time to produce speech sounds which are agreed by other speakers to fall into the classes of speech sounds of their language. No explicit labelling is provided to the learner. By contrast for machines, speech recognition and speech synthesis are studied and trained separately, using labelled speech data.

A structure has been given [1], Fig. 1, which allows the acquisition of speech by machine, asm, and trains recognition and synthesis at the same time from unlabelled training speech. In this synthesis is represented by the operation $f_s(.\mathbf{w}_s)$ which produces synthesised speech s_s from a synthesiser state vector X_s and recognition is represented by the operation $f_r(.\mathbf{w}_r)$ which takes in a speech input s and produces a recogniser state $\mathbf{X}_r$. As shown by the switches in Fig. 1 the recogniser can take its input from the synthesiser s_s and produce a recogniser state $\mathbf{X}_{rs}$ or from human training speech s_t and produce a recogniser state $\mathbf{X}_{rt}$.

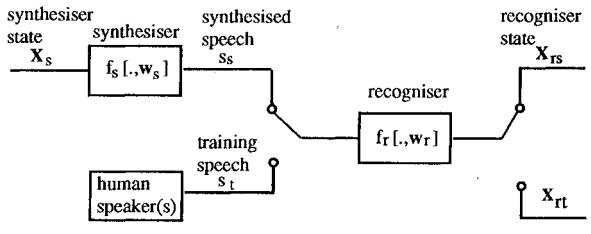

Fig. 1 Structure for asm

The synthesiser state vector is partitioned into two binary vectors, one for synthesiser filter control and one for synthesiser excitation control, and at any instant each binary vector has one element of value 1 and the rest at 0. The recogniser state vector $\mathbf{X}_r$ represents the same level of speech unit as $\mathbf{X}_s$ and has the same dimension as $\mathbf{X}_s$ and when the system is trained it is also of binary form. No limitation is placed on $f_s(.\mathbf{w}_s)$ or $f_r(.\mathbf{w}_r)$ except that when trained by adjustment of the variable parameters $\mathbf{w}_s$ and $\mathbf{w}_r$ they can perform accurately.

It was shown that with a neural net synthesiser controller with weights $\mathbf{w}_s$ and a neural net recogniser with weights $\mathbf{w}_r$, the asm structure can be trained from unlabelled human speech $\mathbf{U}_t$ by the coupled minimisation

$$\min_{\mathbf{X}_s} E_s = \min \sum_{p=1}^{P} \sum_{i=1}^{nor} (X_{rti} - X_{rsi})^2 \qquad (1a)$$

$$\min_{\mathbf{w}_s} E_s = \min \sum_{p=1}^{P} \sum_{i=1}^{nor} (X_{rti} - X_{rsi})^2 \qquad (1b)$$

$$\min_{\mathbf{w}_r} E_r = \min \sum_{p=1}^{P} \sum_{i=1}^{nor} (X_{si} - X_{rti})^2 \qquad (1c)$$

where P is the number of patterns in the training set, nor is the dimension of $\mathbf{X}_r$ (and $\mathbf{X}_s$) and for example X_{rti} is the *ith* element of $\mathbf{X}_{rt}$ at the *pth* pattern.

The structure appears to be quite general in that $\mathbf{X}_s$ can represent any level of speech unit - sub-word, word or upwards. It is also apparent that it is not restricted to the use of neural networks. In this paper it is shown that Hidden Markov Models (HMMs) can also be employed. First it is shown that the asm structure can be decomposed into a canonical form, with benefit. Then the equivalent asm structure using HMMs is given with coupled optimisation equivalent to eqn (1). Finally the equivalence between the neural net and HMM versions of the structure is discussed.

II. DECOMPOSITION OF ASM AND A CANONICAL FORM

A particular example of the asm structure for phoneme acquisition is shown in Fig. 2. Here attention is restricted to training the synthesiser filter only, without loss of generality.

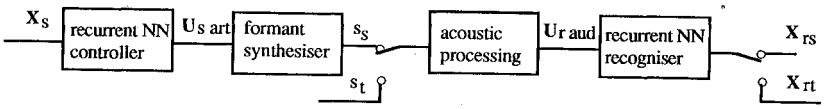

Fig. 2 A particular asm structure

This employs a formant synthesiser and therefore its input, $\mathbf{U}_{s\,art}$, the articulatory control vector, consists of formant frequencies and bandwidths. Also it employs a recurrent net recogniser with an auditory input, $\mathbf{U}_{r\,aud}$, consisting of spectral components, see e.g. [2]. In current speech technology $\mathbf{U}_{s\,art}$ and $\mathbf{U}_{r\,aud}$ are in general different.

Suppose we now introduce a further element - an auditory-articulatory transform $f_{aa}(.\mathbf{w}_{aa})$ as shown in Fig. 3a.

With the switches raised $\mathbf{U}_{r\,aud} = \mathbf{U}_{s\,aud}$ and lowered $\mathbf{U}_{r\,aud} = \mathbf{U}_{t\,aud}$. This has decomposed the asm structure of Fig. 1 into the canonical form shown in Fig. 4. A feature of the asm structure of Fig. 1 is that when trained $f_s(.\mathbf{w}_s)$ and $f_r(.\mathbf{w}_r)$ become inverse operations. In the case of Fig. 4 the introduction of the auditory-articulatory transform means that $f_{s2}(.\mathbf{w}_{s2})$ and $f_{r2}(.\mathbf{w}_{r2})$ become inverse functions, as do $f_{s1}(.\mathbf{w}_{s1})$ and $f_{r1}(.\mathbf{w}_{r1})$, when the structure is trained.

This decomposition has several advantages:

(a) Training is much simplified. The acoustic structure of Fig. 3b is first trained, employing a static neural network for the auditory-articulatory transform $f_{aa}(.\mathbf{w}_{aa})$, and using error-back propagation through the acoustic processing and the synthesiser as described in [3]. Then the outer structure of Fig. 3c is trained. This decomposed training is much simpler than in the case of the complete structure [1].

(b) It appears to be quite general and to apply to any forms of synthesiser or acoustic processing. The decomposition also has biological plausibility in that an auditory-articulatory transform may be learned in humans, allowing us to 'replay conversations in our heads' without hearing or uttering them. Further details of the decomposition and the canonical form for neural network asm are given elsewhere [4], here we employ it for HMM-based asm.

III. AN HMM-BASED ASM STRUCTURE

An HMM-based structure equivalent to that of Fig. 1 is shown in Fig. 5. Again for simplicity and without loss of generality we restrict attention to the synthesiser filter.

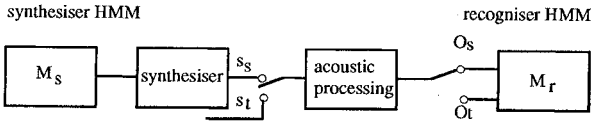

Fig. 5 HMM-based asm structure

Again considerable simplification is produced is by decomposing the structure by the introduction of an auditory-articulatory transform $f_{aa}(.\mathbf{w}_{aa})$, between M_s and the synthesiser in Fig. 5. Assuming its existence, the outer structure becomes that shown in Fig. 6.

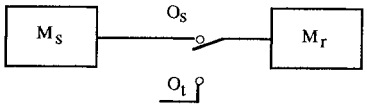

Fig. 6 Outer structure of decomposed model

HMM recognisers are very well known, e.g. [5]. Synthesisers driven by HMMs are also under study, e.g. [6]. However both require labelled training data. The aim of asm is to train without any hand labelling, including symbol order transcription. Again we put no restrictions on the type of synthesiser or acoustic processing employed except that they are appropriate. The synthesiser and recogniser HMMs are required to represent the same level of speech units.

As in the neural net case the HMMs can be controlling the synthesis of, or recognising, any speech unit - sub-word, word or upwards. And the HMMs can be

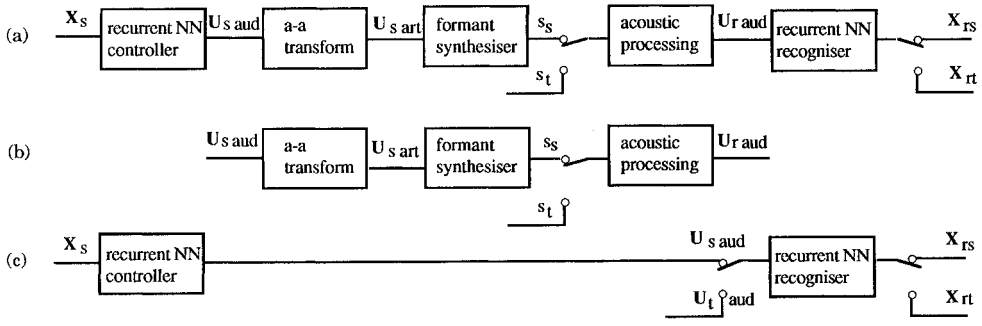

Fig. 3 Decomposition of asm structure (a) complete structure, (b) inner acoustic structure (c) outer structure

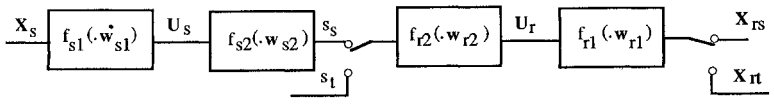

Fig. 4 Canonical form of asm structure

of several types - discrete or continuous, whole word or sub-word, separate models for each unit or composite containing each unit, etc. The asm structure and training method appears to be applicable to any of these.

Concentrating for the moment on HMMs employing three-state phoneme models we see that when trained each synthesiser model will produce all versions of its observations - the sequence of auditory vectors - manifested by that phoneme in the training data, and the recogniser will recognise them. Prior to training these sequences will be different; for the *kth* phoneme, $\mathbf{O}_{sk}$ and $\mathbf{O}_{tk}$ will not overlap, and $\mathbf{O}_{tk}$ will not be emitted with maximum likelihood by its recognition model.

The basic asm equation, for a given set of priors is thus:

$$\max \left(P(\mathbf{O}_{sk} \mid M_{rk})\, P(\mathbf{O}_{tk} \mid M_{rk})\right) \qquad (2)$$

where the first likelihood is maximised by training the synthesiser model M_{sk} and the second by training the recogniser model M_{rk}. As in the case of asm using neural nets via eqn (1), we find that eqn (2) can be solved as a coupled maximisation. First a little on the equations involved.

For an HMM the forward probability or joint probability of being in state i at time t and seeing the first t observation vectors $\mathbf{O}_1^t$ is:

$$P(S(t) = i, \mathbf{O}_1^t \mid M) = \alpha_i(t) \qquad (3)$$

where S(t) is the state of the model at time t. This can be computed recursively as:

$$\alpha_j(t) = \sum_i \alpha_i(t-1)\, a_{ij}\, b_j(\mathbf{o}_t) \qquad (4)$$

where a_{ij} is the transition probability from state i to state j and $b_j(\mathbf{o}_t)$ is the probability of emitting observation vector $\mathbf{o}_t$ from state j. Also the backward probability is:

$$P(\mathbf{o}_{t+1}^T \mid S(t) = i, M) = \beta_i(t) \qquad (5)$$

where T is the duration of the sequence. This can be computed recursively as:

$$\beta_j(t) = \sum_i \beta_i(t+1)\, a_{ij}\, b_j(\mathbf{o}_{t+1}) \qquad (6)$$

From these, the likelihood that the model M emitted the entire sequence $\mathbf{O}_1^T$ is given by:

$$P(\mathbf{O}_1^T \mid M) = \alpha_N(T) \qquad (7)$$

where N is the final, non-emitting state. Also using the Baum-Welch algorithm the HMM can be trained or re-estimated using $\alpha_j(t)$ and $\beta_j(t)$, to maximise $P(\mathbf{O}_1^T \mid M)$ by employing training sequences of phonemes of length T. Both eqn (7) and the Baum-Welch algorithm are required in asm but since the training data $\mathbf{O}_t$, derived from s_t, is not labelled, the method has to provide its own labelling. This is done by the synthesiser, as is seen in the next section.

IV. TRAINING HMM-BASED ASM STRUCTURE

Suppose we assume the HMMs M_s and M_r each have W phoneme HMMs covering the vocabulary of the training data as represented in Fig. 7. (The same applies to other forms of HMMs, such as isolated word models). The HMMs are assumed to have three emitting states, with skips and are placed in loops connecting their final and initial states. The parameters of each HMM are initially randomised, subject to probability constraints.

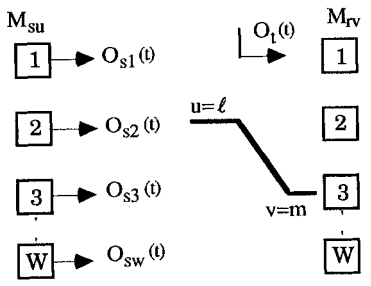

Fig. 7 Individual HMMs

We now decompose eqn (2) as follows, over each phoneme in the training data:

$$\arg\max_{u=\ell, v=m} P(\mathbf{O}_{su1}^{\tau} \mid M_{rv})\, P(\mathbf{O}_{t1}^{\tau} \mid M_{rv}) \qquad (8a)$$

$$\max_{M_{s\ell}} P(\mathbf{O}_{t1}^{\tau} \mid M_{s\ell}) \qquad (8b)$$

$$\max_{M_{rm}} P(\mathbf{O}_{t1}^{\tau} \mid M_{rm}) \qquad (8c)$$

Eqn 8(a) effectively labels the training data and estimates the durations τ of each phoneme in the training data. With all HMMs in their initial states $\mathbf{O}_{t1}$ is applied to all the M_{rv} and $P(\mathbf{O}_{t1} \mid M_{rv})$ is calculated for v=1..W . Then each $\mathbf{O}_{su1}$ is generated for u=1..W and applied to each M_{rv} and each $P(\mathbf{O}_{su1} \mid M_{rv})$ calculated. Then the products of eqn 8(a) are formed and $u=\ell, v=m$ established (this is simplified by finding $\arg\max_{u=u_0} P(\mathbf{O}_{su1} \mid M_{r1})$ and forming its product with $P(\mathbf{O}_{t1} \mid M_{r1})$ for u = 1..W and repeating this for v=2..W and choosing the maximum product). $\mathbf{O}_{t2}$ is then applied, the M_s and M_r allowed a further transition, and ℓ or m recalculated. If ℓ and m are both unchanged then $\tau \geq 2$ and the process is repeated until ℓ or m changes, thereby establishing τ. When ℓ or m changes all the HMMs are re-initialised and the process is restarted from that frame of data. This forms Step 1 of the algorithm. It is applied to all the training data and estimates the labelling and the durations τ of each phoneme in the training data.

In Step 2, eqn (8b) is formed by Baum-Welch re-estimation. We note that for a given ℓ and m, eqn (8b) maximises the first term in eqn (8a) or (2). The process is applied to each phoneme in the training data using the corresponding label ℓ and duration τ found in Step 1.

In Step 3, eqn (8c) is formed by Baum-Welch re-estimation and maximises the second term in eqn (8a) or (2). The process is applied to each phoneme in the training data using the corresponding label m and duration τ.

In Step 4, when the end of training data is reached, all the M_s and M_r of Steps 1-3 are updated.

In Step 5 the process is repeated through the training data until satisfactory maxima of eqn 8 are reached.

We note on completion of training $M_s \rightarrow M_r$, which is a consequence of the canonical form.

For speech synthesis, the state sequences for phonemes can be established by a Viterbi search of the trained M_s and a word lexicon built up.

V. DISCRIMINATIVE TRAINING OF HMM-BASED ASM STRUCTURE AND RELATIONSHIP TO NEURAL NETWORK FORM

Several authors have studied the discriminative training of HMMs and their relationship to neural network recognisers. For example Bridle [7] with his 'alpha-net' gave the theoretical basis of a recurrent neural network architecture with an HMM interpretation; Niles and Silverman [8] gave a neural network representation of individual HMMs as individual recurrent nets and Young [9] in his 'competitive training' method gave a connectionist approach to the discriminative training of HMM's. All employed gradient descent, and showed the identity between the forward probabilities of the HMM and the forward pass of the equivalent neural net, and between the backward probabilities of the HMM and the back-propagation pass of the neural net.

The analogy between the neural network version and the HMM version of asm can be seen by comparing eqns (2) and (8). Further if we make use of Niles and Silverman's results [8] where rather than training by gradient descent to maximise output probabilities, training is done by employing target values, we see that in a composite HMM the resulting gradient descent equations for eqn (2), analogous to eqn (8), become equivalent to the neural net asm equations (1).

VI. CONCLUSIONS

A structure for the acquisition of speech by machine employing HMMs has been presented. It has been seen by analogy with work on HMM-neural network equivalences [8] that the HMM structure and algorithm are equivalent to those of the neural net version of asm. The structure is apparently quite general and can be used with much more elaborate HMMs resulting in improved performance and for different levels of speech such as words or higher. It might be noticed in passing that it is well known that conventionally trained HMM recognisers produce poor quality synthesised speech. The asm work shows as a by-product that this could be improved by introducing an auditory-articulatory transform between the recogniser and a synthesiser.

Computational work is required to demonstrate the performance of the algorithm and this is being pursued. In addition the inclusion of excitation training is required. Some early work [10] provides a basis for this. It is believed that asm gives one principled path for the inclusion of prosody in recognition, and by its inclusion of speaker variation, the improvement of the naturalness of speech synthesis.

VII. REFERENCES

[1] F.Fallside. On the acquisition of speech by machines, asm. Keynote lecture #2, *Eurospeech 91*, Genoa, September 1991 and *Speech Communication.* **11**, 247-260, 1992.

[2] A.J.Robinson and F.Fallside. A recurrent error propagation network speech recognition system, *Computer Speech & Language*, **5**,(3), 259-274, 1991.

[3] F.Fallside. On the acquisition of speech by machine, asm: training the synthesiser controller by error backpropagation. *Proc. IEEE Workshop on Automatic Speech Recognition*, Arden House, 16-19, 1991.

[4] F.Fallside. A neural network structure for the acquisition of speech by machine, asm, employing a canonical decomposition (to be published).

[5] L.R.Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proc. IEEE*, **77**, 2, 257-286, 1989.

[6] A.Falaschi, M.Giustincani and M.Verola. A Hidden Markov Model Approach to speech synthesis. *Proc. NATO ASI Workshop*, Cetraro, Springer Verlag, 187-190, 1990.

[7] J.Bridle. Alpha-nets: a recurrent 'neural' network architecture with a Hidden Markov Model interpretation. *Speech Communication*, **9**, 83-92, 1990.

[8] L.T.Niles and H.F.Silverman. Combining Hidden Markov Model and neural network classifiers. *Proc. ICASSP*, Albuquerque, 417-420, 1990.

[9] S.J.Young. Competitive training: a connectionist approach to the discriminative training of Hidden Markov Models, ibid., 681-684, 1990.

[10] A. Ljolje and F.Fallside. Synthesis of natural sounding pitch contours in isolated utterances using Hidden Markov Models. *IEEE Trans. Acoust. Speech & Signal Processing, ASSP-34*, **5**, 1074-1080, 1986.