

# Unsupervised Learning of Acoustic Sub-word Units

Balakrishnan Varadarajan\* and Sanjeev Khudanpur\* Emmanuel Dupoux

Center for Language and Speech Processing      Laboratoire de Science Cognitive  
Johns Hopkins University                      et Psycholinguistique  
Baltimore, MD 21218                              75005, Paris, France  
{bvarada2, khudanpur}@jhu.edu      emmanuel.dupoux@gmail.com

## Abstract

Accurate unsupervised learning of phonemes of a language directly from speech is demonstrated via an algorithm for joint unsupervised learning of the topology and parameters of a hidden Markov model (HMM); states and short state-sequences through this HMM correspond to the learnt sub-word units. The algorithm, originally proposed for unsupervised learning of allophonic variations within a given phoneme set, has been adapted to learn without any knowledge of the phonemes. An evaluation methodology is also proposed, whereby the state-sequence that aligns to a test utterance is transduced in an automatic manner to a phoneme-sequence and compared to its manual transcription. Over 85% phoneme recognition accuracy is demonstrated for speaker-dependent learning from fluent, large-vocabulary speech.

## 1 Automatic Discovery of Phone(me)s

Statistical models learnt from data are extensively used in modern automatic speech recognition (ASR) systems. Transcribed speech is used to estimate conditional models of the acoustics given a phoneme-sequence. The phonemic pronunciation of words and the *phonemes* of the language, however, are derived almost entirely from linguistic knowledge. In this paper, we investigate whether the phonemes may be learnt automatically from the speech signal.

Automatic learning of phoneme-like units has significant implications for theories of language acquisition in babies, but our considerations here are somewhat more technological. We are interested in developing ASR systems for languages or dialects

for which such linguistic knowledge is scarce or nonexistent, *and* in extending ASR techniques to recognition of signals other than speech, such as manipulative gestures in endoscopic surgery. Hence an algorithm for automatically learning an inventory of intermediate symbolic units—intermediate relative to the acoustic or kinematic signal on one end and the word-sequence or surgical act on the other—is very desirable.

Except for some early work on isolated word/digit recognition (Paliwal and Kulkarni, 1987; Wilpon et al., 1987, etc), not much attention has been paid to automatic derivation of sub-word units from speech, perhaps because pronunciation lexicons are now available<sup>1</sup> in languages of immediate interest. What *has* been investigated is automatically learning allophonic variations of each phoneme due to co-articulation or contextual effects (Takami and Sagayama, 1992; Fukada et al., 1996); the phoneme inventory is usually assumed to be known.

The general idea in allophone learning is to begin with an inventory of only one allophone per phoneme, and incrementally refine the inventory to better fit the speech signal. Typically, each phoneme is modeled by a separate HMM. In early stages of refinement, when very few allophones are available, it is hoped that “similar” allophones of a phoneme will be modeled by shared HMM states, and that subsequent refinement will result in distinct states for different allophones. The key therefore is to devise a scheme for successive refinement of a model shared by many allophones. In the HMM setting, this amounts to simultaneously refining the *topology* and the model *parameters*. A successive state splitting (SSS) algorithm to achieve this was proposed by Takami and Sagayama (1992), and en-

\* This work was partially supported by National Science Foundation Grants Nq IIS-0534359 and OISE-0530118.

<sup>1</sup>See <http://www ldc.upenn.edu/Catalog/byType.jsp>

hanced by Singer and Ostendorf (1996). Improvements in phoneme recognition accuracy using these derived allophonic models over phonemic models were obtained.

In this paper, we investigate directly learning the allophone inventory of a language from speech without recourse to its phoneme set. We begin with a one-state HMM for all speech sounds and modify the SSS algorithm to successively learn the topology and parameters of HMMs with even larger numbers of states. States sequences through this HMM are expected to correspond to allophones. The most likely *state-sequence* for a speech segment is interpreted as an “allophonic labeling” of that speech by the learnt model. Performance is measured by mapping the resultant state-sequence to phonemes.

One contribution of this paper is a significant improvement in the efficacy of the SSS algorithm as described in Section 2. It is based on observing that the improvement in the goodness of fit by up to two consecutive splits of any of the current HMM states can be evaluated *concurrently and efficiently*. Choosing the best subset of splits from among these is then cast as a constrained knapsack problem, to which an efficient solution is devised. Another contribution of this paper is a method to evaluate the accuracy of the resulting “allophonic labeling,” as described in Section 3. It is demonstrated that if a small amount of phonetically transcribed speech is used to learn a Markov (bigram) model of state-sequences that arise from each phone, an evaluation tool results with which we may measure phone recognition accuracy, even though the HMM labels the speech signal not with phonemes but merely a state-sequence. Section 4 presents experimental results, where the performance accuracies with different learning setups are tabulated. We also see how as little as 5 minutes of speech is adequate for learning the acoustic units.

## 2 An Improved and Fast SSS Algorithm

The improvement of the SSS algorithm of Takami and Sagayama (1992), renamed ML-SSS by Singer and Ostendorf (1996), proceeds roughly as follows.

1. Model all the speech<sup>2</sup> using a 1-state HMM with a *diagonal-covariance* Gaussian. ( $N=1$ )

<sup>2</sup>Note that the original application of SSS was for learning

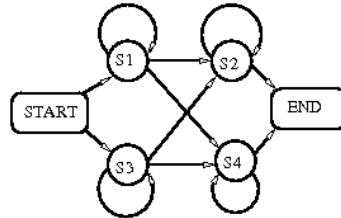


Figure 1: Modified four-way split of a state  $s$ .

2. For each HMM state  $s$ , compute the gain in log-likelihood (LL) of the speech by either a contextual or a temporal split of  $s$  into two states  $s_1$  and  $s_2$ . Among the  $N$  states, select and split the one that yields the most gain in LL.
3. If the gain is above a threshold, retain the split and set  $N = N + 1$ ; furthermore, if  $N$  is less than desired, re-estimate all parameters of the new HMM, and go to Step 2.

Note that the key computational steps are the for-loop of Step 2 and the re-estimation of Step 3.

**Modifications to the ML-SSS Algorithm:** We made the following modifications that are favorable in terms of greater speed and larger search space, thereby yielding a gain in likelihood that is potentially greater than the original ML-SSS.

1. Model all the speech using a 1-state HMM with a *full-covariance* Gaussian density. Set  $N = 1$ .
2. Simultaneously replace each state  $s$  of the HMM with the 4-state topology shown in Figure 1, yielding a  $4N$ -state HMM. If the state  $s$  had parameters  $(\mu_s, \Sigma_s)$ , then means of its 4-state replacement are  $\mu_{s_1} = \mu_s - \delta = \mu_{s_4}$  and  $\mu_{s_2} = \mu_s + \delta = \mu_{s_3}$ , with  $\delta = \epsilon \lambda^* v^*$ , where  $\lambda^*$  and  $v^*$  are the principal eigenvalue and eigenvector of  $\Sigma_s$  and  $0 < \epsilon \ll 1$  is typically 0.2.
3. Re-estimate all parameters of this (overgrown) HMM. Gather the Gaussian sufficient statistics for each of the  $4N$  states from the last pass of re-estimation: the state occupancy  $\pi_{s_i}$ . The sample mean  $\mu_{s_i}$ , and sample covariance  $\Sigma_{s_i}$ .
4. Each quartet of states (see Figure 1) that resulted from the same original state  $s$  can be

used to estimate the allophonic variations of a phoneme; hence the phrase “all the speech” meant all the speech corresponding *separately* to each phoneme. Here it really means all the speech.

merged back in different ways to produce 3, 2 or 1 HMM states. There are 6 ways to end up with 3 states, and 7 to end up with 2 states. Retain for further consideration the 4 state split of  $s$ , the best merge back to 3 states among the 6 ways, the best merge back to 2 states among the 7 ways, and the merge back to 1 state.

5. Reduce the number of states from  $4N$  to  $N + \Delta$  by *optimally*<sup>3</sup> merging back quartets that cause the least loss in log-likelihood of the speech.
6. Set  $N = N + \Delta$ . If  $N$  is less than the desired HMM size, retrain the HMM and go to Step 2.

Observe that the 4-state split of Figure 1 permits a slight look-ahead in our scheme in the sense that the goodness of a contextual or temporal split of two different states can be compared in the same iteration with two consecutive splits of a single state. Also, the split/merge statistics for a state are gathered in our modified SSS assuming that the other states have already been split, which facilitates consideration of concurrent state splitting. If  $s_1, \dots, s_m$  are merged into  $\tilde{s}$ , the loss of log-likelihood in Step 4 is:

$$\frac{d}{2} \sum_{i=1}^m \pi_{s_i} \log |\Sigma_{\tilde{s}}| - \frac{d}{2} \sum_{i=1}^m \pi_{s_i} \log |\Sigma_{s_i}|, \quad (1)$$

where  $\Sigma_{\tilde{s}} = \frac{\sum_{i=1}^m \pi_{s_i} (\Sigma_{s_i} + \mu_{s_i} \mu'_{s_i})}{\sum_{i=1}^m \pi_{s_i}} - \mu_{\tilde{s}} \mu'_{\tilde{s}}$ .

Finally, in selecting the best  $\Delta$  states to add to the HMM, we consider many more ways of splitting the  $N$  original states than SSS does. E.g. going up from  $N = 6$  to  $N + \Delta = 9$  HMM states could be achieved by a 4-way split of a single state, a 3-way split of one state and 2-way of another, or a 2-way split of three distinct states; all of them are explored in the process of merging from  $4N = 24$  down to 9 states. Yet, like SSS, no original state  $s$  is permitted to merge with another original state  $s'$ . This latter restriction leads to an  $O(N^5)$  algorithm for finding the best states to merge down<sup>4</sup>. Details of the algorithm are omitted for the sake of brevity.

In summary, our modified ML-SSS algorithm can leap-frog by  $\Delta$  states at a time, e.g.  $\Delta = \alpha N$ , compared to the standard algorithm, and it has the benefit of some lookahead to avoid greediness.

<sup>3</sup>This entails solving a constrained knapsack problem.

<sup>4</sup>This is a restricted version of the 0-1 knapsack problem.

### 3 Evaluating the Goodness of the Labels

The HMM learnt in Section 2 is capable of assigning state-labels to speech via the Viterbi algorithm. Evaluating whether these labels are linguistically meaningful requires *interpreting* the labels in terms of phonemes. We do so as follows.

Some phonetically transcribed speech is labeled with the learnt HMM, and the label sequences corresponding to each phone segment are extracted. Since the HMM was learnt from unlabeled speech, the labels and short label-sequences usually correspond to allophones, not phonemes. Therefore, for each *triphone*, i.e. each phone tagged with its left- and right-phone context, a simple *bigram* model of label sequences is estimated. An unweighted “phone loop” that accepts all phone sequences is created, and composed with these bigram models to create a label-to-phone transducer capable of mapping HMM label sequences to phone sequences.

Finally, the test speech (not used for HMM learning, nor for estimating the bigram model) is treated as having been “generated” by a source-channel model in which the label-to-phone transducer is the source—generating an HMM state-sequence—and the Gaussian densities of the learnt HMM states constitute the channel—taking the HMM state-sequence as the channel input and generating the observed speech signal as the output. Standard Viterbi decoding determines the most likely phone sequence for the test speech, and phone accuracy is measured by comparison with the manual phonetic transcription.

## 4 Experimental Results

### 4.1 Impact of the Modified State Splitting

The ML-SSS procedure estimates  $2N$  different  $N+1$ -state HMMs to grow from  $N$  to  $N+1$  states. Our procedure estimates *one*  $4N$  state HMM to grow to  $N+\Delta$ , making it hugely faster for large  $N$ .

Table 1 compares the log-likelihood of the training speech for ML-SSS and our procedure. The results validate our modifications, demonstrating that at least in the regimes feasible for ML-SSS, there is no loss (in fact a tiny gain) in fitting the speech data, and a big gain in computational effort<sup>5</sup>.

<sup>5</sup>ML-SSS with  $\Delta=1$  was impractical beyond  $N=22$ .

# of states	SSS ( $\Delta = 1$ )	$\Delta = 3$	$\Delta = N$
8	-7.14	-7.13	-7.13
10	-7.08	-7.06	-7.06
22	-6.78	-6.76	N/A
40	N/A	-6.23	-6.20

Table 1: Aggressive state splitting does not cause any degradation in log-likelihood relative to ML-SSS.

## 4.2 Unsupervised Learning of Sub-word Units

We used about 30 minutes of phonetically transcribed Japanese speech from *one* speaker<sup>6</sup> provided by Maekawa (2003) for our unsupervised learning experiments. The speech was segmented via silence detection into 800 utterances, which were further partitioned into a 24-minute training set (80%) and 6-minute test set (20%).

Our first experiment was to learn an HMM from the training speech using our modified ML-SSS procedure; we tried  $N = 22, 70$  and 376. For each  $N$ , we then labeled the training speech using the learnt HMM, used the phonetic transcription of the training speech to estimate label-bigram models for each triphone, and built the label-to-phone transducer as described in Section 3. We also investigated (i) using only 5 minutes of training speech to learn the HMM, but still labeling and using all 24 minutes to build the label-to-phone transducer, and (ii) setting aside 5 minutes of training speech to learn the transducer and using the rest to learn the HMM. For each learnt HMM+transducer pair, we phonetically labeled the test speech.

The results in the first column of Table 2 suggest that the sub-word units learnt by the HMM are indeed interpretable as phones. The second column suggests that a small amount of speech (5 minutes) may be adequate to learn these units consistently. The third column indicates that learning how to map the learnt (allophonic) units to phones requires relatively more transcribed speech.

## 4.3 Inspecting the Learnt Sub-word Units

The most frequent 3-, 4- and 5-state sequences in the automatically labeled speech consistently matched particular phones in specific articulatory contexts, as

<sup>6</sup>We heeded advice from the literature indicating that automatic methods model gross channel- and speaker-differences before capturing differences between speech sounds.

HMM	24 min	5 min	19 min
label-to-phone	24 min	24 min	5 min
27 states	71.4%	70.9%	60.2%
70 states	84.4%	84.7%	75.8%
376 states	87.2%	86.8%	76.6%

Table 2: Phone recognition accuracy for different HMM sizes ( $N$ ), and with different amounts of speech used to learn the HMM labeler and the label-to-phone transducer.

shown below, i.e. the HMM learns allophones.

HMM labels	L-contxt	Phone	R-contxt
11, 28, 32	<i>vowel</i>	t	[e a o]
15, 17, 2	[g k]	[u o]	[*]
3, 17, 2	[k t g d]	a	[k t g d]
31, 5, 13, 5	<i>vowel</i>	[s sj sy]	<i>vowel</i>
17, 2, 31, 11	[g t k d]	[a o]	[t k]
3, 30, 22, 34	[*]	a	<i>silence</i>
6, 24, 8, 15, 22	[*]	o	<i>silence</i>
4, 3, 17, 2, 21	[k t]	a	[k t]
4, 17, 24, 2, 31	[s sy z] [t d]	o	[t d] [s sy z]

For instance, the label sequence 3, 17, 2, corresponds to an “a” surrounded by stop consonants {t, d, k, g}; further restricting the sequence to 4, 3, 17, 2, 21, results in restricting the context to the unvoiced stops {t, k}. That such clusters are learnt without knowledge of phones is remarkable.

## References

- T. Fukada, M. Bacchiani, K. K. Paliwal, and Y. Sagisaka. 1996. Speech recognition based on acoustically derived segment units. In *ICSLP*, pages 1077–1080.
- K. Maekawa. 2003. Corpus of spontaneous japanese: its design and evaluation. In *ISCA/IEEE Workshop on Spontaneous Speech Processing and Recognition*.
- K. K. Paliwal and A. M. Kulkarni. 1987. Segmentation and labeling using vector quantization and its application in isolated word recognition. *Journal of the Acoustical Society of India*, 15:102–110.
- H. Singer and M. Ostendorf. 1996. Maximum likelihood successive state splitting. In *ICASSP*, pages 601–604.
- J. Takami and S. Sagayama. 1992. A successive state splitting algorithm for efficient allophone modeling. In *ICASSP*, pages 573–576.
- J. G. Wilpon, B. H. Juang, and L. R. Rabiner. 1987. An investigation on the use of acoustic sub-word units for automatic speech recognition. In *ICASSP*, pages 821–824.