A Discriminative Segmental Speech Model and its Application to Hungarian Number Recognition

László Tóth,[1] András Kocsor[2] and Kornél Kovács[3]

[1,2,3]Research Group on Artificial Intelligence of the Hungarian Academy of Sciences and of the University of Szeged
H-6720 Szeged, Aradi vértanúk tere 1., Hungary
{tothl, kocsor, coree}@inf.u-szeged.hu

Abstract. This paper presents a stochastic segmental speech recognizer that models the a posteriori probabilities directly. The main issues concerning the system are segmental phoneme classification, utterance-level aggregation and the pruning of the search space. For phoneme classification artificial neural networks and support vector machines are applied. Phonemic segmentation and utterance-level aggregation is performed with the aid of anti-phoneme modeling. At the phoneme level the system convincingly outperforms the HMM system trained on the same corpus, while at the word level it attains the performance of the HMM system trained without embedded training.

1 Introduction

The currently most popular stochastic approach to automatic speech recognition statistically models the joint distribution $P(W, A)$ of the acoustic observations A and the possible transcriptions W. During recognition an incoming signal is identified as the transcription with the maximum a posteriori probability, so the result is

$$W^* = \arg\max_W P(W|A) = \arg\max_W P(A|W)P(W), \tag{1}$$

where the latter, decomposed form is derived using Bayes' formula.We will call those models that use this decomposed form and work with $P(A|W)$ "generative".

We know about very few systems that model $P(W|A)$ directly. Some authors refer to these as "recognition"[6], or "perception"[5] (vs. "production") models. Owing to the lack of a commonly accepted name we will call these models "discriminative", in accordance with general pattern recognition terminology. Since this may be misleading (generative models can be trained discriminatively as well), we should stress here that we use this naming for the *models* and not necessarily their *training technique*.

This paper describes a segment-based recognizer built on discriminatively trained segmental classifiers. Section 2 discusses the segmental framework in general. Section 3 then describes the phoneme classifier and related issues, while Section 4 presents our results. The paper closes with concluding remarks and planned future developments.

2 Segment-Based Recognition

In the following we suppose that a speech signal A is given as a series of frame-based observation vectors $A = a_1 a_2 ... a_T$, while the possible transcriptions W are given as series of phonemic labels $W = w_1 w_2 ... w_n$. We will model the conditional probability $P(W|A)$ in a decomposed form, as we want to keep our system general enough for continuous speech recognition as well. In our framework $P(W|A)$ is decomposed as

$$P(W|A) = \prod_i P(w_i|A) = \prod_i P(w_i|A_i), \tag{2}$$

where the first equation covers the assumption that the phonemes are independent (we presume phonetic correlation to be modeled by an independent language model), and the second equation reflects the assumption that the identity of a particular phoneme w_i depends only on a particular segment $A_i = a_j a_{j+1} a_{j+t}$. The phonemic probabilities $P(w_i|A_i)$ can then be trained on a manually segmented and labeled corpus. The learning algorithms that are used to model $P(w_i|A_i)$ will be called the "phoneme classifier", which will be discussed in the next section.

Equation (2) implicitly assumes that we know the phonetic segmentation of the signal. However, since automatic segmentation cannot be done reliably, we have to evaluate many possible segmentations S during recognition. This means that we work with $P(W,S|A)$ from which S has to be finally removed by marginalization. Formally,

$$P(W|A) = \sum_S P(W,S|A) = \sum_S P(W|S,A)P(S|A) \approx \max_S P(W|S,A)P(S|A). \tag{3}$$

For a given S, $P(W|S,A)$ can be calculated using equation (2). The more problematic issue is with $P(S|A)$. From a practical viewpoint it gives a weighting of the phoneme models that normalizes the different segmentation paths. One might try a heuristic "aggregation function" to combine the classifier outputs, but a bad strategy could lead to errors like the preference of short or long words. The most popular solution for avoiding the problems associated with $P(S|A)$ is to run a frame-based (e.g. HMM) recognizer, and re-score only the N best paths by the segmental phoneme models[11]. We, however, wanted to model $P(S|A)$ with discriminative classifiers, for which we trained segmental probabilities $P(s_i|A_i)$. In this two-class training the phonemes of the manually labeled corpus acted as positive examples for the class "phoneme", while (quasi-)randomly cut pieces of the database served as examples of the "anti-phoneme", whose class thus covers any segment that is a part of or a composite of some phonemic segments. One motivation for approximating $P(S|A)$ from the phoneme/anti-phoneme probabilities was that it made it possible to train $P(s_i|A_i)$ and $P(w_i|A_i)$ on the same features. This allowed us to unite the two classifiers into one, which considerably decreased the computational costs.

The simplest way to approximate $P(S|A)$ from the values $P(s_i|A_i)$ is

$$P(S|A) = \prod_{s_i \in S} P(s_i|A_i). \tag{4}$$

Unfortunately, this formula does not guarantee proper normalization between different segmentations. For this all segments should be considered, like in

$$P(S|A) = \prod_{s_i \in S} P(s_i|A_i) \prod_{s_j \in \overline{S}} (1 - P(s_j|A_j)), \tag{5}$$

where $\overline{S}$ denotes the set of *all other* segments that occur in *any* other segmentation. However, the space of all segments is prohibitively large. So we approximated the second product in (5) by considering only those elements of $\overline{S}$ that are "near-misses" of the elements of S. More precisely, for a given segment $s \in S$ we utilized the anti-phoneme probability of the two nearest segments in $\overline{S}$ that overlap either boundaries of s. To get the best performance these probabilities had to be raised to empirically chosen powers (indicating that these values represent also those segments not utilized explicitly).

As the third important issue concerning segmental recognition, we stress that the approximation of the sum with the best segmentation in (3) turns the summation into a search problem. In contrast to an HMM system where the probabilities belonging to different state sequences can be evaluated efficiently by dynamic programming, in segmental models the probability of a segment cannot be simply composed from the probabilities of "sub-segments" (i.e. frames), but need a call to the classifier. Thus for acceptable execution speed the effective pruning of the space of possible segmentations is crucial. Our system performs a depth-first search to find the best segmentation of an utterance. When a leaf is reached in the search space, its probability value serves as a threshold to avoid traversing less promising segmentation paths. In this way, on the average, the system can find the N best segmentations quite efficiently.

Finally, the number of possible segmentations can be reduced drastically by the application of a signal processing algorithm that segments the utterance based on local changes in the spectrum. When choosing the parameters for such an algorithm, one should bear in mind that insertion errors only increase the search space, but deletion errors have the risk of completely misrecognizing the utterance in question.

3 Discriminative Phoneme Classification

3.1 Segmental Features

Although there are many sophisticated segmental models offered in the literature (e.g. [1]), we used a simple technique similar to that of the SUMMIT system[3]. At the frame level the speech signals were represented by their critical-band log-energies, and the averages of the 24 critical-band log-energies of the segment thirds (divided in a 1-2-1 ratio) were used as segmental features for phoneme classification. The advantage of this method is that it needs only trifling additional calculations following the computation of the frame-based features. Moreover, it returns the same number of segmental features independent of the segment length, which was a prerequisite for the classifiers used.

Besides phoneme classification, we also needed features that discriminate phonemes from anti-phonemes. We used the variances of the features along the segments to filter out candidates that contain boundaries inside them, and the derivatives of the features at the boundaries to remove candidates with improbable start and end-points. These segmental features were calculated only on 4 wide frequency bands, as it proved sufficient.

A special segmental feature is the duration of the phoneme. We consider it especially important for languages like Hungarian where phonemic duration can play a discriminative role. As our preliminary experiments found duration to be useful indeed, it was employed as a segmental feature in all our experiments. Thus, including duration, 77 features altogether were used to represent the segments.

3.2 Feature Space Transformations

A special problem with discriminative segmental models is that since the observations are from a continuous space, contextual variability cannot simply be addressed by triphone models, as is usual with generative modeling. Furthermore, since the number of features increases the computational cost of most classifiers non-linearly, it should be kept as low as possible. For these reasons we attach great importance to feature space transformation methods. These methods may aid classification performance and can also reduce the dimensionality of the data. Linear discriminant analysis (LDA), principal component analysis (PCA) and independent component analysis (ICA) are the traditional (linear) transformation techniques [2][8]. Recently the non-linear version of these linear transformations have become a popular research topic in statistical learning theory. We performed experiments applying the so-called "Kernel non-linearization idea"[10][8] on LDA. Rather than going into mathematical details, we demonstrate the effect of this transformation on two artificially generated data sets, both consisting of two classes. Figure 1 shows that in both cases the otherwise interweaving classes become separable by one straight line after the transformation (the sets were encircled only for the ease of visualization).

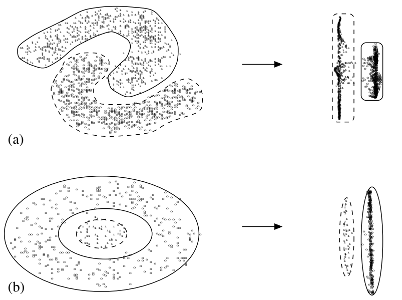

Fig. 1. The effect of Kernel-LDA on two point sets, both consisting of two classes

3.3 Classifiers

It is known that artificial neural networks (ANN), under proper conditions, can be used to approximate a posteriori probabilities[7]. In a previous study [4] we also found ANNs to be the best for phoneme classification. In the experiments below "ANN" means three-layer MLPs trained with back-propagation. The number of hidden neurons was 150 in the phoneme classification, and 50 in the phoneme/anti-phoneme classification tests.

For the classification task we also experimented with a promising new technique called support vector machines (SVM). Owing to the lack of space we refer the interested reader to Vapnik[10] for an overview on SVM. In all the experiments with SVM a second-order polynomial kernel function was applied.

4 Experimental Results

We evaluated our system on a small corpus, where the training set consisted of 20, while the test set of 6 talkers pronouncing 52-52 Hungarian numbers. The recordings were of reasonably good quality, sampled and quantized at 22050 Hz, 16-bit. The whole database was manually segmented and labeled. Because of the restricted domain, the corpus contained only 28 different phonemic labels.

For a comparison, an HMM system was also trained on the same corpus using monophone models (the corpus is too small to train triphones). The description of the HMM recognizer can be found at Szarvas[9].

4.1 Phoneme-Level Results

Table 1 shows the segmental classification errors. In the case of the phoneme classification (28 classes) we have a comparative result from the HMM which shows that the segmental discriminative models give significantly better results. In addition, one can notice that the classifiers attained the same performance after LDA and K-LDA, in spite of the transformations considerably reducing the number of features. Similar observations hold for the phoneme plus anti-phoneme (29 classes) and phoneme/anti-phoneme classification tasks (in the latter case no transformation was applied, as there were only two classes).

		No transf. (77 feat.)	LDA (27 feat.)	K-LDA (27 feat.)
28 phonemes	HMM	9.34%	—	—
	ANN	7.78%	7.81%	5.79%
	SVM	5.81%	5.12%	4.59%
28 ph. + antiph.	ANN	6.78%	6.87%	6.54%
	SVM	7.90%	6.14%	5.89%
phoneme/ antiph.	ANN	6.92%	—	—
	SVM	5.10%	—	—

Table 1. Segmental classification error rates

4.2 Word-Level Results

Table 4 shows the error rates on the word level (all experiments were performed with ANN classifiers). The first column shows the error rate when the recognizer examined all possible segments that can be composed from 5-frame chunks of the signal. The result is comparable with the performance of the HMM (without embedded training), but the runtime was over an order of magnitude worse.

The result in the second column was attained when the recognizer used only the segments found by an algorithm that looked for local changes in the spectrum. On the average this algorithm cut a phoneme into only 1.84 pieces. Runtime with this search space reduction was close to real-time, but unfortunately the error rate became more than double. Visual inspection showed that almost all new errors were caused by deletion errors in the automatic segmentation.

Finally, we ran an experiment substituting the manual segmentation in place of the automatic segmentation mentioned above. Surprisingly, we got worse results than with the 5-frame "fake-segmentation", which indicates that the manual segmentation in many cases does not coincide with the boundaries suggested by the acoustic features.

From this we concluded that a version of the expectation-maximization algorithm (that is iterative “recognize-and-retrain” loops) could significantly improve the system's performance. The fact that the HMM performed much better with embedded training than with training on the manual segmentation also reinforces this assumption.

Segmental Model			HMM	
5-frame segm.	Autom. segm.	Manual segm.	No embed. tr.	Embedded tr.
1.28%	3.20%	1.92%	1.60%	0.32%

Table 2. Word error rates

5 Conclusion

From our results (and the similar ones found in the literature) we conclude that although segmental models can quite easily outperform HMM on the phoneme level, this gain can be easily lost on the utterance level. In our case it means that a better aggregation strategy, that is a better approximation of $P(S|A)$ must be found – preferably one that can be trained discriminatively at the utterance level. Also, to reach acceptable execution speed we have to look for some better pruning methods, since the spectrally-based segmentation algorithm proved unreliable, and doing a full search (using the 5-frame fake segments) is very slow. Implementation of an iterative training algorithm also promises much improvement. Finally, this far we used only a fixed vocabulary, so an interesting question is the integration with a probabilistic language model, which quite probably needs some different technique than in the case of an HMM system.

References

1. Fukada, T., Sagisaka, Y. and Paliwal, K. K., Model Parameter Estimation for Mixture Density Polynomial Segment Models, *Proc. of ICASSP'97,* pp. 1403-1406, Munich, Germany, 1997.
2. Fukunaga, K., *Statistical Pattern Recognition,* New York: Academic Press, 1989.
3. Halberstadt, A. K., Heterogeneous Measurements and Multiple Classifiers for Speech Recognition, *Ph.D. Thesis, Dep. Electrical Engineering and Computer Science, MIT,* 1998.
4. Kocsor, A., Tóth, L., Kuba, A. Jr., Kovács, K., Jelasity, M., Gyimóthy, T. and Csirik, J., A Comparative Study of Several Feature Transformation and Learning Methods for Phoneme Classification, accepted for publication in the *International Journal of Speech Technology*.
5. Mariani, J., Gauvain, J. L., Lamel, L., Comments on “Towards increasing speech recognition error rates” by H. Bourlard, H. Hermansky, and N. Morgan, *Speech Communication,* 18 (1996), pp. 249-252.
6. Morgan, N., Bourlard, H., Greenberg, S., Hermansky, H., Stochastic Perceptual Auditory-Event-Based Models for Speech Recognition, *Proc. of ICSLP'94,* pp. 1943-1946, 1994.
7. Richard, M. D. and Lippmann, R. P., Neural network classifiers estimate Bayesian a posteriori probabilities, *Neural Computation,* 3(4):461:483, 1991.
8. Schölkopf, B., Smola, A. And Müller, K. -R., Nonlinear Component Analysis as a Kernel Eigenvalue Problem, *Neural Computation,* Vol. 10(5), 1998.
9. Szarvas, M., Mihajlik, P., Fegyó, T. and Tatai, P., Automatic Recognition of Hungarian: Theory and Practice, accepted for publication in the *International Journal of Speech Technology*.
10. Vapnik, V. N., *Statistical Learning Theory,* John Wiley & Sons Inc., 1998.
11. Zavaliagkos, G., Zhao, J., Schwartz, R. and Makhoul, J., A Hybrid Segmental Neural Net/Hidden Markov Model System for Continuous Speech Recognition, *IEEE Trans. Speech and Audio Proc.*, Vol. 2, No. 1, Part II, January 1994.