TOWARDS ARTICULATORY SPEECH RECOGNITION: LEARNING SMOOTH MAPS TO RECOVER ARTICULATOR INFORMATION

*Sam Roweis**
Computation and Neural Systems
California Institute of Technology

Abeer Alwan
Department of Electrical Engineering
University of California, Los Angeles

Abstract

We present a novel method for recovering articulator movements from speech acoustics based on a constrained form [9] of a hidden Markov model. The model attempts to explain sequences of high dimensional data using smooth and slow trajectories in a latent variable space. The key insight is that this continuity constraint when applied to speech helps to solve the "ill-posed" problem of acoustic to articulatory mapping. By working with *sequences* of spectra rather than looking only at individual spectra, it is possible to choose between competing articulatory configurations for any given spectrum by selecting the configuration "closest" to those at nearby times. We present results of applying this algorithm to recover articulator movements from acoustics using data from the Wisconsin X-ray microbeam project [3]. We find that the recovered traces are highly correlated with the measured articulator movements under a single linear transform. Such recovered traces have the potential to be used for speech recognition, an application we are currently investigating.

1 Introduction

A potent objection to speech recognition techniques as they exist today is the lack of speech-specific knowledge in them. The existing models could be (and are) equally well used to identify machine noises or DNA sequences. By incorporating prior information about the nature of speech, the problem can be heavily constrained. Such regularization is essential to achieving more robust and accurate performance than current approaches permit.

One way to view automatic speech recognition is as a code-breaking problem [1]. There is an *unknown message* (say a sequence of phonemes) which has been *encoded* into a pressure versus time waveform. Our goal is to *decode* the waveform and recover the message. Two important sources of prior information are commonly used to aid us in this decoding. First, we know a lot about the *set of possible messages* because languages have very strong syntactic structure; this motivates work on language modeling for use in recognition systems. Second, we know a lot about the *receiver of the code* through studies of human perception; this allows us to preprocess the waveform by emphasizing perceptually relevant features.[1] Such studies motivate short-time spectral analysis as an almost universal preprocessing step in recognizers. However, we also know a lot about the *producer of the code* from studies of speech production; the ultimate aim of the present work is to use speech production models to improve speech recognition systems.

2 Methods

In particular we draw upon *continuity*, *low-frequency energy* and *dimensionality reduction* as essential constraints from speech production models. Because humans speak using articulator movements that are for the most part slow and smooth,[2] the complicated acoustic signals which we observe should ultimately be explainable by slowly varying latent variables (the articulator movements). Furthermore, since the articulatory system has a limited number of degrees of freedom (estimates vary from 3 to 10), the number of such independent variables required should be small. Such explicability constraints are an important feature of speech signals and are not obviously true of, for example, DNA sequences.

In this paper we present a novel approach for recovering information about articulator movements from acoustics alone. We apply a constrained form [9] of a hidden Markov model to learn low-dimensional trajectories which explain observed high-dimensional time series. This is similar in spirit to other dimensionality reduction techniques such as Kohonen mapping or principle component analysis but contains a crucial new element: we require the learned

*roweis@cns.caltech.edu; Mail Code 139-74, California Institute of Technology, Pasadena CA. 91125 U.S.A.

[1] Many preprocessors *discard* certain information about the waveform. Continued intelligibility by humans is one (but by no means the only) way to ensure that such preprocessing does not render the unknown message unrecoverable. In other words if we know that a certain manipulation of the acoustic signal leaves it still intelligible to humans then we can be sure that it has not destroyed *essential* information. However it may certainly have destroyed *useful* information.

[2] While individual articulators may at times make abrupt movements in general there is little power above 20 Hz in measured traces. See for example [2].

trajectories to only vary slowly and smoothly even though the original (acoustic) data may have high frequency components or abrupt spectral changes. The algorithm assigns emission probabilities over a high-dimensional output space to fixed cells in a low-dimensional map. These probabilities are learned such that smooth paths in the low-dimensional map generate sequences like those in the observed data, or equivalently, so that the observed data have high likelihood with only slow and smooth paths in the map. The model admits both noise in the output observations as well as the possibility for more than one cell in the low-dimensional map to have similar output probabilities.

By way of illustration, consider playing the following game: divide a sheet of paper into several contiguous non-overlapping regions which between them cover it entirely. In each region inscribe an integer, allowing numbers to be repeated in different regions. Now place a pencil on the sheet and move it around, reading out (in order) the numbers in the regions through which it passes. Add some *noise* to the observation process so that occasionally an incorrect number is reported in the list. The goal of the game is to reconstruct the configuration of regions on the sheet from only such an ordered list of noisy numbers.

How does this model relate to speech processing? The low-dimensional map in which we are learning trajectories (the "sheet" in our game) represents an abstract "articulatory" space (for example articulator positions and voicing condition). The output probabilities for each cell in the sheet represent the sound type which is produced when the articulators are in the configuration specified by the cell's location. (In our game this corresponded to the choice of number in each region.) From this viewpoint, playing the game outlined above is akin to pursuing an *acoustic to articulatory mapping*: we perform short-time spectral analysis on an incoming acoustic signal, classify each short-time spectrum into one of a finite number of categories, and then attempt to reconstruct the trajectories in the articulatory space based only on the observed sequences of pattern numbers.

By allowing symbols to be repeated in the map, we are recognizing that there may be several *different* articulatory configurations all of which produce very *similar* spectral patterns. This one-to-many aspect of the acoustic to articulatory mapping has led many researchers to declare the inverse mapping problem to be impossible or at least ill-posed. However, this objection ignores the crucial continuity property that articulators possess. Thus, while we may not be able to specify a *single* articulatory configuration given a *single* spectrum, we may indeed be able to reconstruct articulatory *traces* from *sequences* of spectra. This use of temporal information should allow better recovery of articulator information than simple "inverse-lookup" procedures used in the past [8].

3 Details of the algorithm

The basic algorithm used to learn the low-dimensional trajectories is a variant of the standard hidden Markov model learning algorithm. The key difference is that the state transition matrix is *precomputed* and fixed throughout the learning. The precomputation is achieved by first identifying each state in the Markov model with a cell in a low-dimensional packing of space (we have used hexagonal and cubic packings). The transition matrix is computed by selecting some self-transition probability for each state which will control the typical speed of trajectories through the map. All remaining probability is then distributed equally amongst the states which correspond to neighbours in the cell packing. The transition probability to non-neighbour states is fixed to zero. In this way, all legal state sequences in the model correspond to slow and smooth paths in some low-dimensional space.

Given these fixed transition probabilities, the output emission probabilities for each cell are learned using the standard Baum-Welch updating procedure. It is possible to use both continuous valued and discrete outputs. For the experiments reported below we have used only discrete valued outputs. Once the learning of the output probabilities has converged, any particular observed data sequence can then be *decoded* (using the equivalent of the Viterbi algorithm) to discover its corresponding state trajectory in the low-dimensional map. Since there are a finite number of cells (states) in the map and since they are located at the grid-points of a particular packing, the raw recovered trajectories will jump from grid-point to grid-point. A continuous low-dimensional trajectory is produced from these recovered state sequence by interpolation using a kernel whose power spectrum matches the average power spectrum of the observed data in each dimension.

4 Results

We present results on synthetic data (Figs. 1 & 2), which include noise in the observations as well as repeated numbers in the true maps, indicating that the algorithm can reliably recover smooth maps. The synthetic data were produced by making random walks (with self transition probability 10%) in the true map of Figure 1 to produce sequences of output numbers. Notice that the true map contains duplicates of the numbers 6 and 1. These output sequences were then corrupted with noise by replacing 15% of the symbols with a random symbol. These noisy sequences were then used to learn the map shown in Figure 2. Although the learned map shown here has the same number of states and topology (hexagonal packing) as the true map, results are similar if the learned map has more states or cubic packing.

In addition, we apply the algorithm to real speech data from the Wisconsin speech production database [3]. This database contains simultaneously recorded acoustic and articulatory data. The acoustic data are sampled at roughly 21 kHz and the articulatory data consist of roughly 150 Hz sampling of 8 articulator positions on the midsaggital plane. The acoustic data were preprocessed by computing 12 mel-frequency cepstral coefficients based on 23.5 ms windows at a frame rate of 6.9 ms (designed to match the articulatory sampling). These cepstral coefficients were then vector-quantized using a codebook of 64 symbols. The codebook was trained using a batch version of the *k-means* algorithm. We then applied the constrained Markov model with state spaces of various dimensions from 1 to 10 to these processed acoustic data.

The trajectories that were recovered (Figs. 3-6) in the map space are highly correlated[3] with the actual measured articulator traces (after being appropriately linearly transformed). In this sense, the algorithm performs a kind of acoustic to articulatory mapping. The original mapping is *unsupervised*: articulatory data are not used to train the system. However, in order to evaluate the algorithm, after it had been run, articulatory data were used to compute the best *single* linear transformation between all the learned trajectories in the map and all the actual observed articulator movements. This single transform can then be applied to any individual trace to generate a recovered articulator trajectory. In our experiments, all utterances were used to fit the transform, but the results are similar if a random subset of half are left out of the fitting.

It is important to emphasize that only a very limited set of sounds were used in these tests with speech data. In particular we used 56 noiseless utterances of vowels, consonant-vowel-consonant triples (/s/V/d/) and vowel-consonant-vowel triples (/uh/C/a/) all from a single male speaker. Nonetheless, our results are encouraging from the point of view of being able to recover a simple linear transformation of articulator information from acoustics alone. Notice that it is *impossible* to recover anything better than an orthogonal transformation (rotation plus axis aligned scaling) of the true information since the coordinate axes and measurement units used to specify the articulator movements are both arbitrary.

Most previous work in acoustic to articulatory mapping (for example [4,8]) has focused on entirely supervised methods which do not directly incorporate the continuity constraint. One notable exception is the related unsupervised algorithm of Hogden [5] which was an early inspiration for many of the ideas in this work.

[3]The average correlation coefficient was at least 0.9 for maps of dimension 4 or greater.

5 Future Work

Our motivation for pursuing an acoustic to articulatory mapping is the belief that speech recognition may be significantly more tractable in the articulatory domain.[4] We are currently in the process of using this recovered articulatory information as input to a recognition system, to evaluate if such knowledge increases performance. Earlier studies ([6],[7]) using similar data but supervised mappings have shown promising results in this direction. We are also investigating the recovery of articulatory information from a larger range of speech sounds and from continuous speech, as well as investigating the degradation of the mapping as acoustic noise is added.

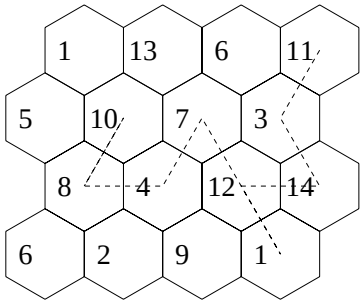

Figure 1: Original synthetic map and a portion of a sample trajectory. Notice that numbers 6 and 1 are repeated. Trajectories were random walks with 15% probability of outputting a random symbol instead of the region number.

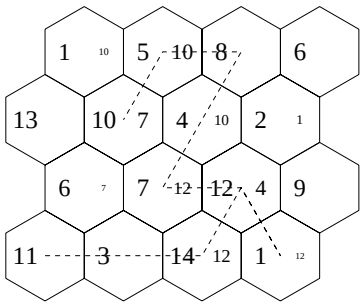

Figure 2: Learned map (derived from a 16 state HMM) and recovered trajectory. Font size indicates symbol probability. The HMM was trained on a sequence of 500 noisy output numbers generated from the map above. Note the key feature: contiguous cells in the original map tend to be contiguous in the learned one; hence trajectories in the learned map remain smooth.

[4]The main reason for this belief is the observation (for example see [4]) that variability, which is a major source of the difficulty in recognition, may be easier to identify and account for in the articulatory domain. That is, certain aspects of articulation are reliably repeated while other aspects are highly variable across repetitions of the same utterance. This variability manifests itself as generalized acoustic variability which plagues traditional spectral-feature based recognizers. However if recognition were performed in the articulatory domain, a system might learn to rely only on the reliable articulation patterns, thereby achieving more robust recognition.

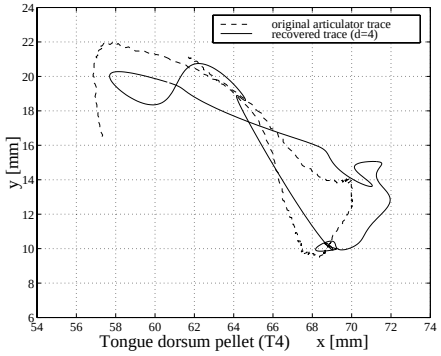

Figure 3: A measured trace of actual speech articulator data and its reconstruction. The horizontal and vertical axes are the x and y coordinates of the pellet. For this example, a 4-dimensional map containing 1296 cells and 64 output symbols was learned. The recovered trace shown here is a linear transformation of a trajectory learned from only acoustic, not articulator data. A single linear transformation was used to project all the map trajectories onto the actual articulator traces, as opposed to a different transform for each pair.

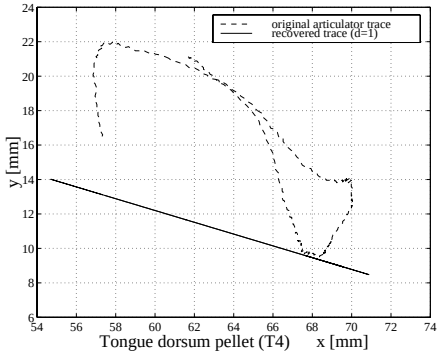

Figure 4: Similar reconstruction for a 1-dimensional map.

6 Acknowledgments

Sam Roweis is grateful for many fruitful and detailed discussions with John Hopfield and his group at Caltech as well as for the support of Abeer Alwan and her group at UCLA. He would also like to acknowledge John Hodgen whose work and comments form an important basis for these ideas. Roweis is supported in part by the Center for Neuromorphic Systems Engineering as a part of the National Science Foundation Engineering Research Center Program under grant EEC-9402726 and by the Natural Sciences and Engineering Research Council of Canada under an NSERC 1967 Award.

7 References

[1] Jelinek, F. (1992). Speech recognition as a code breaking process. Research report 5, CLSP, Johns Hopkins University.

[2] Muller, E. & McLeod, G. (1982). Perioral biomechanics and its relation to labial motor control. J.Acoust.Soc.Am. 78.

[3] Westbury, J. (1994). X-ray microbeam speech production database user's handbook. University of Wisconsin, Madison.

[4] Papcun G. et al. (1992). Inferring articulation and recognizing gestures from acoustics with a neural network trained on x-ray microbeam data. J.Acoust.Soc.Am. 92(2) pp. 688-700

[5] Hogden, J. (1996). Improving on hidden markov models: An articulatorily constrained, maximum likelihood approach to speech recognition and speech coding (unclassified LA-UR-96-3945). Los Alamos, NM: Los Alamos National Laboratory.

[6] Zacks, J. & Thomas, T. (1996). A new neural network for articulatory speech recognition and its application to vowel identification. Unpublished manuscript.

[7] Zlokarnik, I. (1995). Adding articulatory features to acoustic features for automatic speech recognition. J.Acoust.Soc.Am 97(5 p2).

[8] Schroeter, J. & Sondhi, M. (1994). Techniques for estimating vocal tract shapes from the speech signal. IEEE Trans. Speech and Audio Proc. 2(1 p2) pp. 133-150

[9] Roweis, S. (1997) Low-dimensional maps for explaining sequences of high dimensional data. CNS Technical Report, California Institute of Technology.

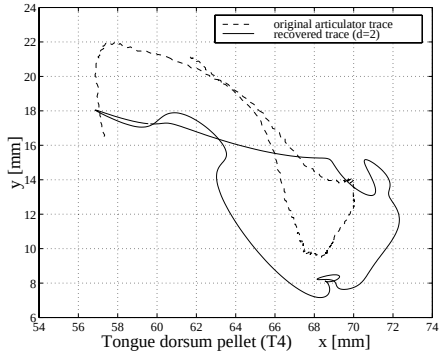

Figure 5: Similar reconstruction for a 2-dimensional map.

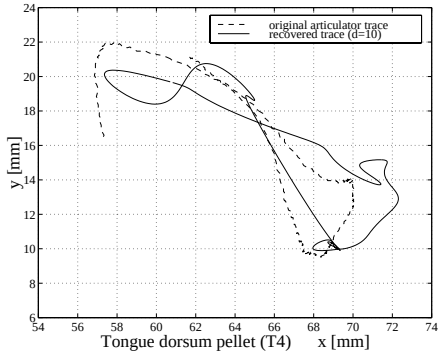

Figure 6: Similar reconstruction for a 10-dimensional map. Notice that there is little improvement after 4 dimensions.