INTERSPEECH 2009 BRIGHTON

Preliminary Inversion Mapping Results with a New EMA Corpus

Korin Richmond.

Centre for Speech Technology Research,
University of Edinburgh, UK
korin@cstr.ed.ac.uk

Abstract

In this paper, we apply our inversion mapping method, the trajectory mixture density network (TMDN), to a new corpus of articulatory data, recorded with a Carstens AG500 electromagnetic articulograph. This new data set, `mngu0`, is relatively large and phonetically rich, among other beneficial characteristics. We obtain good results, with a root mean square (RMS) error of only 0.99mm. This compares very well with our previous lowest result of 1.54mm RMS error for equivalent coils of the MOCHA `fsew0` EMA data. We interpret this as showing the `mngu0` data set is potentially more consistent than the `fsew0` data set, and is very useful for research which calls for articulatory trajectory data. It also supports our view that the TMDN is very much suited to the inversion mapping problem.
Index Terms: acoustic-articulatory, inversion mapping, neural network.

10.21437/Interspeech.2009-724

1. Introduction

There is a large body of research which has investigated ways to exploit an articulatory representation of speech to improve both speech technology and understanding. Various articulography methods have been employed to provide useful data to support this work, such as electropalatography (EPG), X-ray microbeam cinematography, ultrasound and MRI. The MOCHA `fsew0` electromagnetic articulography (EMA) corpus has been particularly useful and has supported research in areas such as ASR [1], articulatory-acoustic synthesis [2, 3], speech synthesis and the inversion mapping [4].

We have previously worked extensively on the inversion mapping problem. To perform the inversion mapping, we aim to take an acoustic signal and estimate the sequence of articulatory movements which produced it. This would be useful, for example, as a source of articulatory features for several speech technology applications (in addition to those mentioned above) such as low bit-rate speech coding, visual speech synthesis or facial animation, speech communication augmented by speech synthesis [5] and speech training software.

The method we have been developing is based on a particular type of artificial neural network (ANN): the mixture density network (MDN). The MDN allows us to model a probability distribution over variables in the articulatory domain conditioned on the acoustics features. Supplementing the static articulatory features with their delta and deltadelta features allows us to form a statistical trajectory model for articulatory movements, which we have termed the trajectory MDN (TMDN). Similar to many others, we have frequently used the MOCHA `fsew0` dataset as training and test data [6, 7, 8, 9]. The TMDN has proved very well suited to the inversion mapping problem, and the results we have obtained, in terms of root mean square (RMS) error expressed in millimetres, are lower than any other comparable study we have seen so far [6].

Although the TMDN performs well, it is hard to know how far it is from the optimal performance possible. In theory, there are three sources for the error we observe: error attributable to inaccurate modelling, error attributable to inconsistencies in the articulatory data recordings, and "residual" error resulting purely from intrinsic variability of articulatory movements. The question is, how much of the error exhibited by the TMDN in [6] is due to the first two of these, and which is hence open to improvement? This paper aims to investigate this question. First, we consider evidence to suggest there may be some inconsistency in the `fsew0` corpus. Then, we introduce a new corpus, `mngu0`, which has certain possible advantages. To evaluate this, we train a TMDN to perform the inversion mapping on `mngu0`, and consider what the results might tell us.

2. Evidence for inconsistency

In this section we consider evidence hinting at some degree of inconsistency in the `fsew0` EMA data set. One clear source of inconsistency is introduced where a coil becomes detached and needs to be re-attached during the recording session. Unfortunately, it is not possible to re-attach a sensor coil with exactly the same position and orientation. It is also possible the movement necessary to re-attach a coil could result in a shift in the position of the speaker's head relative to the EMA helmet, which potentially affects the accuracy of all coils. The recording log for `fsew0` indicates the velum coil needed to be re-attached at recording index 125, and the middle tongue ("TB") coil was re-attached at file recording index 284.

Looking at the EMA data itself in plots like Figure 1, we observe what may be evidence of inconsistency. In this figure, we have overlaid the sampled position of the velum recorded throughout multiple utterances. These utterances comprise two groups of contiguously recorded files: group 1 (black) for the sixteen files 070−085; group 2 (grey) for the 11 files 102−112. We observe the movement of the velum appears to be very regular; it is constrained to movement in a slight arc. However, we furthermore note that for the two groups of files we have selected, while they exhibit the same pattern of velum movement, it appears as though the coordinate system has been somehow shifted or rotated between the two groups. The cause of this variation is unknown. These two groups of files both came before the point during the recording session at which it is reported the velum coil became detached. Therefore, it is conceivable that it is unrelated to that event. Potential causes might be movement of the speaker's head within the AG200 helmet between the two groups, or differences introduced by the head-movement correction algorithm. This relied on tracking two reference coils, and if either of these coils became inaccurate for some reason (such as movement off the midline axis of the transmitter coils, or movement with the skin of the bridge of the nose), then the coordinate system for the rest of the coils would

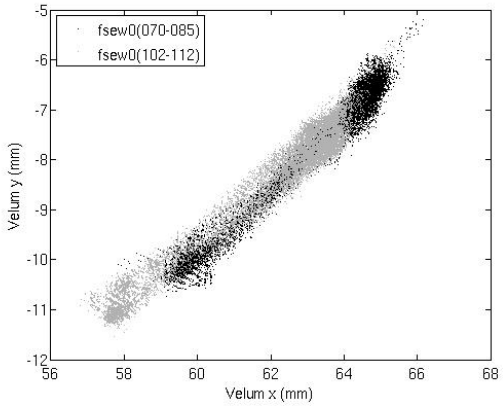

Figure 1: Overlaid samples of velum x- and y-coordinates for multiple files in the MOCHA fsew0 data set. Two groups of contiguous files are shown: one group (black) shows velum positions for files $070 - 085$, the other (grey) shows the velum for files $102 - 112$

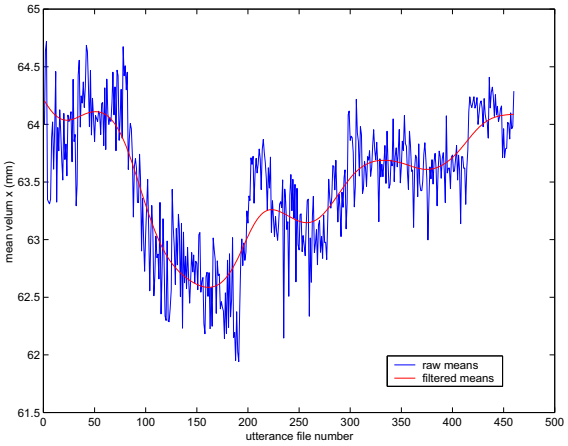

Figure 2: A plot of mean velum x-coordinate calculated for each utterance from the database for speaker `fsew0` in recording order. This diagram also shows trends captured by lowpass filtering the file means.

be affected. Thus, it is possible the positions of the other coils are affected in a way similar to the translation we observe in the patterns of velum movement in Figure 1.

Figure 2 gives another perspective on potential inconsistency present in `fsew0`. This plot shows the mean velum x-coordinate (y-axis) calculated for each of the 460 files in the data set (along the x-axis) [10]. We would expect the mean velum position to vary "randomly" in accordance with phonetic content of each file. For example, for an utterance containing more nasal stops, we would anticipate the mean velum position to be lower, and vice-versa. In Figure 2, we observe such random variation across neighbouring files. However, we also observe more global trends in the position of the velum.

In [10], we used a simple method of normalisation in an attempt to reduce the prominence of these trends. Taking each x- and y-coordinate for each coil separately, the method basically consists of calculating the mean coordinate for each file, lowpass filtering this to identify trends over time (see the smooth line overlaid on Figure 2), and then using this mean-trend as part of z-score normalisation. Inversion mapping experiments in [10] showed this normalisation reduced RMS error between the measured articulatory trajectories and the estimated ones. This suggests the global trends are indeed symptomatic of inconsistency in the EMA data.

3. New EMA data set (`mngu0`)

We have available an alternative EMA data set: `mngu0`, recorded using a Cartsens AG500 electromagnetic articulograph at Ludwig-Maximilians-Universität München[1]. This data set consists of over 2,000 utterances recorded from a single speaker over two consecutive days. For the session of the first day, over 1,200 utterances were recorded with EMA sensors attached to the speaker's upper lip (UL), lower lip (LL), lower incisor (JAW), tongue tip (T1), tongue blade (T2) and tongue dorsum (T3), plus coils used for head-movement correction. On the second day, a coil was placed on the velum (V), with only two coils on the tongue. Around 800 utterances were recorded with this configuration. Coils were attached in the midsagittal plane. The prompts for the speaker were selected from newspaper text, using a text-selection algorithm designed for building `Multisyn` [11] unit-selection voices for `Festival`. This algorithm aimed to maximise coverage of context-specific diphones in as few sentences as possible.

In addition to providing a large and phonetically rich source of EMA data, the `mngu0` corpus offers other advantages. First, none of the sensor coils used became detached during recording, immediately increasing the chances of a consistent data set. Using the AG500 conferred several advantages. Unlike the preceding 2D AG200, this system tracks sensor coils in 3D space, and with two angles of rotation. Among other benefits, this means the speaker's head is free to move, which increases comfort. It also obviates the problem with the AG200 of inaccuracy being introduced when a sensor moves off the midline plane of the transmitter coils (although sensor coil tracking in the AG500 is a non-linear optimization problem which may bring other uncertainties). Finally, the `mgnu0` data was recorded with speech synthesis in mind, so care was taken to ensure good audio quality and a professional actor was employed. Unlike for MOCHA, EPG was not used, and so the impact on the speech from the articulography equipment is only very negligible.

We plan to release this dataset in the near future. In addition to the raw EMA and audio data, we aim to release the processed versions we have used in our experiments. We shall likewise release phonetic labelling, based on the `Combilex` lexicon and created using forced alignment. The intention is to enable other researchers to conduct experiments using exactly the same data, in order to facilitate direct comparison between different approaches. Furthermore, we have collected additional data from the same speaker, such as MRI scans and video of the mouth area, which we shall also release to form a collection of articulatory-acoustic data for a single speaker.

4. Inversion experiment

In this preliminary experiment, we aim to use the `mngu0` data set described in Section 3 to train the TMDN system [6] to perform the acoustic-articulatory inversion mapping.

[1] In collaboration with Phil Hoole

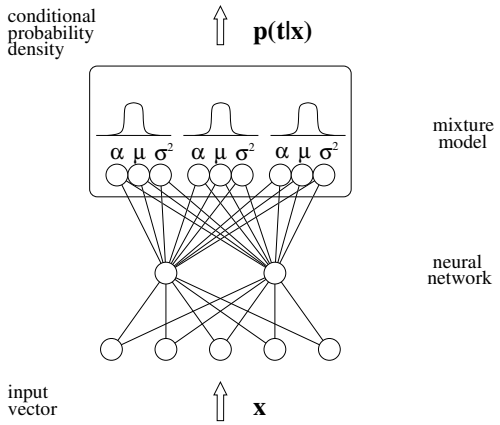

Figure 3: The mixture density network we use combines an MLP and GMM

.

4.1. TMDN

Due to space constraints, we can only give a high level introduction to the TMDN. For a more explicit description, the reader is referred to [6].

At heart of our inversion mapping model is the mixture density network (MDN). In the most general sense, the MDN can be considered as combining a trainable regression function (typically a non-linear regressor such as an artificial neural network) with a probability density function. In our work, we have been using a multilayer perceptron (MLP) as a trainable non-linear regressor and a Gaussian mixture model (GMM). An illustration is shown in Figure 3. The role of the MLP is to take an input vector in one domain ($\mathbf{x}$, acoustic features in this case) and map to the control parameters (priors, means and variances) of the pdf over the domain of the target parameters ($\mathbf{t}$, the EMA positions). In this way, the MDN offers a model of probability density over the target domain conditioned on the input domain, $p(\mathbf{t}|\mathbf{x})$. Training consists of updating the MLP weights to optimize an error function, defined as the negative log likelihood of the target data. As in standard MLP training, the error at the MLP outputs (which have specialized activation functions) may be calculated and backpropagated to find the gradient of the error function with respect to the weights. Thus, standard non-linear optimization algorithms may be used to train the MDN.

Since, the MDN gives us a model of conditional probability density, it is trivial to augment the target features with derived delta and deltadelta features. Once trained, we may then input the sequence of acoustic feature vectors for an utterance and get as output a sequence of pdfs over the static articulatory features and their delta and deltadeltas. We may then apply a maximum likelihood parameter generation algorithm (MLPG)[12] to this sequence of pdfs in order to obtain a single, most probable trajectory which optimizes the constraints between the distributions of static, delta and deltadelta features. In the case of a sequence of pdfs containing a single Gaussian mixture component, this optimum is the solution of a set of linear equations. When multiple mixture components are used, an EM-based algorithm described in [12] is applied.

4.2. Data processing

Part 1 of the `mngu0` dataset was used for the experiment presented here. 6 coils were attached to the speaker's articulators in the midsagittal plane: 3 on the tongue, one on the lower incisor and one each on the upper and lower lips. Coil movement in the axis orthogonal to the midsagittal plane was very small. To match previous results using the `fsew0` corpus, we used only the movements of the coils in the midsagittal plane. So, the articulatory data used for this experiment comprised 12 channels of EMA data at a sampling frequency of 200Hz.

The corresponding audio data was converted to frequency-warped LSFs of order 40 plus a gain value. These were derived from the spectral envelope estimated with STRAIGHT, which was calculated at a 5msec frame shift to match the sample rate of the articulatory data[2]. Finally, both EMA and LSF features vectors were z-score normalised, by subtracting their respective global mean and dividing by four times the standard deviation.

Three subsets were created from a total set of 1,263 utterances: a training set of 1,137 utterances comprising all files apart from those with an index number ending in '0'; a validation set of 63 utterances comprising the half of the held out utterances with an odd integer preceding the final '0'; and a test set with the remaining 63 utterances.

training stage	# subsets	# epochs each subset	stage total weight updates
1	20	5	100
2	10	10	100
3	5	20	100
4	2	50	100
5	1	2600	2600
weight update grand total =			3000

Table 1: Incremental training schedule used to train the TMDN with various subsets of the training set.

4.3. Training procedure

For the experiment presented here, we have used a context window of 10 acoustic frames as input. This context window was constructed, however, by alternately selecting only every other acoustic frame. Thus, given the 5msec shift of the acoustic feature frames, there was a 90msec time difference between the acoustic frames at the left and right edge of the context window. With 41 features in each acoustic vector, the size of the MDN input layer was 410 units. We used 100 units in the hidden layer, each with a $tanh$ activation function. We trained separate MDNs with output pdfs containing either 1, 2 or 4 mixture components. Furthermore, each of the 12 articulatory channels were trained in a separate MDN, meaning a total of 36 MDNs were trained.

The Scaled Conjugate Gradients optimization algorithm was used to train the MDN weights. Network training was conventional in that we used a held-out validation set to guard against over-fitting. However, to accelerate training, we used a schedule of updating the network weights according to error calculated on varying subsets of the training set. This schedule is presented in Table 1. For example, in the first stage, we split the training set into 20 subsets and for each of these calculated the error and updated the weights 5 times. This means that the weights were updated a total of 100 times in stage 1, but that only one twentieth of the training data was used for any one update. In stage 2, the number of separate subsets was decreased, and the number of weight updates made was increased. This pattern was repeated for the following stages, until in stage 5, where all training data was used to update the weights a total of 2,600 times. Calculating the error function and its gradient for only one twentieth of the training set is certainly faster, al-

[2]This processed data set is identically to that used in [5]

Channel	TMDN 1 mix	TMDN 2 mix	TMDN 4 mix	opt # mixes
T3_x	1.25	1.28	1.22	4
T3_y	1.74	1.63	1.57	4
T2_x	1.41	1.40	1.34	4
T2_y	1.38	1.24	1.24	2
T1_x	1.41	1.39	1.36	4
T1_y	1.30	1.27	1.28	2
JAW_x	0.57	0.57	0.57	2
JAW_y	0.75	0.74	0.75	2
UL_x	0.32	0.32	0.32	2
UL_y	0.46	0.46	0.49	2
LL_x	0.66	0.64	0.64	2
LL_y	1.16	1.11	1.18	2

Table 2: Inversion results for `mngu0`: RMS error (mm) between TMDNs with 1, 2 or 4 mixture components. Average (min) RMSE=0.99mm for all coils.

.

Channel	static lpfilt	TMDN 1 mix	TMDN 2 mix	TMDN 4 mix	opt # mixes
upper lip x	0.90	0.90	0.90	0.91	1
upper lip y	1.06	1.05	1.03	1.06	2
lower lip x	1.11	1.10	1.10	1.12	1
lower lip y	2.31	2.27	2.20	2.22	2
lower incisor x	0.84	0.82	0.80	0.81	2
lower incisor y	1.05	1.03	1.04	1.03	1
tongue tip x	2.14	2.12	2.09	2.10	2
tongue tip y	2.12	2.08	1.98	1.94	4
tongue body x	1.99	1.96	1.97	1.98	1
tongue body y	1.80	1.76	1.73	1.78	2
tongue dorsum x	1.88	1.85	1.81	1.83	2
tongue dorsum y	1.92	1.89	1.85	1.88	2
velum x	0.36	0.35	0.35	0.35	2
velum y	0.37	0.37	0.36	0.37	2

Table 3: Comparable TMDN results for `fsew0` from [6]. As well as RMS error (mm) for TMDNs with 1, 2 and 4 mixture components, results are shown for a low-pass filtered static feature mean sequence (equivalent to smoothed output of standard MLP). Discounting the velum coil, the average (min) RMSE is 1.54mm.

.

though it does mean that the error "landscape" being explored by the optimization algorithm is not necessarily consistent between updates. However, this can be viewed as similar to the case of applying simulated annealing in function optimization, with an analogous schedule of "cooling".

5. Results

The results of applying the TMDNs trained with `mngu0` to the held-out test set are given in Table 2. Comparable results previously reported for the `fsew0` data in [6] are in Table 3.

We find the TMDNs using `mngu0` performed very well, significantly better than previous results obtained using `fsew0`: RMS error of 0.99mm instead of 1.54mm for equivalent EMA coils. A plausible explanation is that the `mngu0` corpus may indeed feature a lower level of inconsistency than `fsew0`.

However, we cannot be sure what degree of inconsistency might be present in the `mngu0` corpus. Specifically, we are unfortunately not able to judge how much of the error we still observe is attributable to inadequacies in the model, how much is due to potential inconsistency in the articulatory data, and how much is residual error due simply to inherent variability in articulatory movements. Hopefully, other researchers will apply different techniques to the same data set. This will provide reference points to help decide how good a model the TMDN is for the inversion mapping. At the moment, we can merely posit that the `mngu0` appears to be a reasonably good data set.

6. Conclusions

We have introduced a new corpus of EMA data and used it to train our TMDN inversion mapping method. Performance was very good, even better than previously obtained with MOCHA `fsew0` data. Our results indicate the new `mngu0` corpus is a good resource, and that the TMDN is an accurate model of the inversion mapping.

7. Acknowledgements

K. Richmond is currently supported by EPSRC grant EP/E027741/1. This work has made use of the resources provided by the Edinburgh Compute and Data Facility (ECDF) (http://www.ecdf.ed.ac.uk/). The ECDF is partially supported by the eDIKT initiative (http://www.edikt.org.uk).

8. References

[1] S. King, J. Frankel, K. Livescu, E. McDermott, K. Richmond, and M. Wester, "Speech production knowledge in automatic speech recognition," *Journal of the Acoustical Society of America*, vol. 121, no. 2, pp. 723–742, February 2007.

[2] C. Kello and D. Plaut, "A neural network model of the articulatory-acoustic forward mapping trained on recordings of articulatory parameters," *J. Acoust. Soc. Am.*, vol. 116, pp. 2354–2364, 2004.

[3] T. Toda, A. W. Black, and K. Tokuda, "Statistical mapping between articulatory movements and acoustic spectrum using a Gaussian mixture model," *Speech Communication*, vol. 50, no. 3, pp. 215–227, Mar. 2008.

[4] C. Qin and M. . Carreira-Perpin, "A comparison of acoustic features for articulatory inversion," in *Proc. Interspeech*, 2007, pp. 2469–2472.

[5] Z. Ling, K. Richmond, J. Yamagishi, and R. Wang, "Integrating articulatory features into HMM-based parametric speech synthesis," *IEEE Transactions on Audio, Speech and Language Processing*, 2009, accepted for publication.

[6] K. Richmond, "Trajectory mixture density networks with multiple mixtures for acoustic-articulatory inversion," in *Advances in Nonlinear Speech Processing, International Conference on Non-Linear Speech Processing, NOLISP 2007*, ser. Lecture Notes in Computer Science, M. Chetouani, A. Hussain, B. Gas, M. Milgram, and J.-L. Zarader, Eds., vol. 4885. Springer-Verlag Berlin Heidelberg, Dec. 2007, pp. 263–272.

[7] ——, "A multitask learning perspective on acoustic-articulatory inversion," in *Proc. Interspeech*, Antwerp, Belgium, aug 2007.

[8] ——, "A trajectory mixture density network for the acoustic-articulatory inversion mapping," in *Proc. Interspeech*, Pittsburgh, USA, September 2006.

[9] K. Richmond, S. King, and P. Taylor, "Modelling the uncertainty in recovering articulation from acoustics," *Computer Speech and Language*, vol. 17, pp. 153–172, 2003.

[10] K. Richmond, "Estimating articulatory parameters from the acoustic speech signal," Ph.D. dissertation, The Centre for Speech Technology Research, Edinburgh University, 2002.

[11] R. A. J. Clark, K. Richmond, and S. King, "Multisyn: Open-domain unit selection for the Festival speech synthesis system," *Speech Communication*, vol. 49, no. 4, pp. 317–330, 2007.

[12] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, Istanbul, Turkey, June 2000, pp. 1315–1318.