# A Hybrid ANN/DBN Approach to Articulatory Feature Recognition

*Joe Frankel*,* *Simon King*†

Centre for Speech Technology Research
The University of Edinburgh
joe@cstr.ed.ac.uk

## Abstract

Artificial neural networks (ANN) have proven to be well suited to the task of articulatory feature (AF) recognition. Previous studies have taken a cascaded approach where separate ANNs are trained for each feature group, making the assumption that features are statistically independent. We address this by using ANNs to provide virtual evidence to a dynamic Bayesian network (DBN). This gives a hybrid ANN/DBN model and allows modelling of inter-feature dependencies. We demonstrate significant increases in AF recognition accuracy from modelling dependencies between features, and present the results of embedded training experiments in which a set of asynchronous feature changes are learned. Furthermore, we report on the application of a Viterbi training scheme in which we alternate between realigning the AF training labels and retraining the ANNs.

## 1. Introduction

We first give a general motivation for our research, then describe the context and focus of the work presented in this paper.

### 1.1. Motivation

This paper describes work which is part of an ongoing project to build an automatic speech recognition (ASR) system where a set of articulatory features (AF), rather than phones, provide the internal representation mediating between the word string and the acoustic observation sequence. The primary motivation for this approach is to move away from the limitations of using phones, i.e. the "beads-on-a-string" paradigm [1]. Generating words as concatenations of phone models makes it difficult to model the variation that is present in spontaneous, conversational speech. Conventional systems use context-dependent phone models to deal with this variation. We argue that AFs offer a representation which can be used to derive a compact and efficient model of the contextual and pronunciation variation encountered by a speaker-independent recognition system.

### 1.2. Context of the current study

Previous studies using AFs for recognition have typically reverted to phones at some level. In the word recognition system we are currently implementing, we avoid re-introducing the "beads-on-a-string" paradigm by describing words as sequences of feature values. We choose to work with a dynamic Bayesian network (see Section 4) framework for the following reasons:

- DBNs make it possible to model the dependencies between feature streams.
- DBNs provide a unified framework in which to integrate the components of a feature-based recognition system.
- Inference and estimation algorithms are available for whole model classes, so prototyping novel models and topology modification is simple (using GMTK [2]).

Figure 1 shows a single time-slice of our system in graphical model notation. In our model, feature states, rather than sub-phone states, generate observations. For each of the 6 features (see Table 1 below), a set of templates are defined, each of which specifies a sequence of 1 or more feature values. A word is generated by specifying templates to dictate the behaviour of each of the features (i.e. the sequence of values that feature must pass through for this word). Variation due to pronunciation or context is in terms of features and is encoded by the templates: for any given feature, multiple templates may be associated with a word. This dependence is modelled probabilistically.

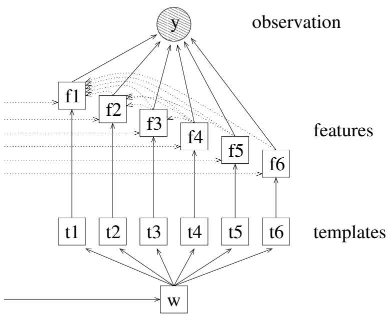


Figure 1: *Graph depicting one time-slice of the AF-based recognition system we are currently implementing. Square/round, shaded/unshaded denote discrete/continuous, observed/hidden nodes respectively, arrows denote dependencies and for clarity inter-feature dependencies are shown with dotted lines.*

Figure 2 illustrates template-based modelling of pronunciation variation. Two possible manner templates for the word "four" are shown, each with different prior probability, which produce different alignments with the observation frames.

One of the central aims of this research is to use embedded training to automatically learn pronunciation variation in terms of articulatory features. To do so, we require an informed initialization of the various components of the model, in particular the observation process which, as shown by the top half of Figure 1, consists of an articulatory feature recognizer. Previous work on AF recognition has included deriving a set of inter-feature dependencies [3], and using embedded training to bypass some of the limitations of training on feature labels derived from time-aligned phone labels [4]. In this paper, we further refine our modelling of the observation process through the use of artificial neural networks (ANN).

*Supported by EPSRC grant GR/S21281/01
†Supported by EPSRC Advanced Fellowship GR/T04649/01

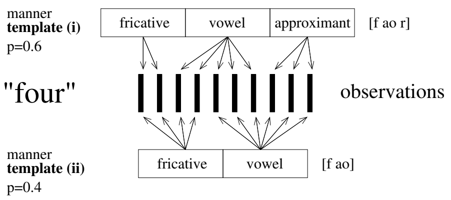


Figure 2: *Illustration of pronunciation modelling using templates, showing two possible manner feature strategies for generating the word "four".*

### 1.3. Focus of the current study

A number of studies, for example [3, 5, 6], have shown that ANNs are capable of recognizing articulatory features with high accuracy. Furthermore, ANNs have the benefit of being computationally cheap during recognition, and given their discriminative nature, produce greater separation in hypothesis likelihoods. Previous studies have employed a cascaded approach in which separate ANNs are trained for each feature group. This gives the advantage of robust classification [5], though makes the assumption that features are independent of each other.

We have found the performance of our best DBN system, which uses a Gaussian mixture model (GMM) observation process, to be comparable to the performance of ANNs [4]. Each of these methods have advantages: the DBNs can model inter-feature dependencies, whilst the ANNs are discriminatively trained. In this paper, we build on the strengths of each of these two models, and present a hybrid ANN/DBN approach.

## 2. Data

Experimental work uses a subset of the Numbers Corpus [7] (OGI Numbers), a collection of naturally spoken numbers collected at the Center for Spoken Language Understanding (CSLU). The utterances include isolated digit strings, continuous digit strings, and ordinal/cardinal numbers, and have all been orthographically and phonetically transcribed following the CSLU Labelling Conventions [8]. The train and test sets consist of a little over 6 and 2 hours of recorded speech respectively. In all experiments, the acoustic waveforms are parameterized as 12 MFCCs and energy with $1^{st}$ and $2^{nd}$ derivatives.

| feature | values | cardinality |
|---|---|---|
| manner | approximant, fricative, nasal, stop, vowel, silence | 6 |
| place | labiodental, dental, alveolar, velar, high, mid, low, silence | 8 |
| voicing | voiced, voiceless, silence | 3 |
| rounding | rounded, unrounded, nil, silence | 4 |
| front-back | front, central, back, nil, silence | 5 |
| static | static, dynamic, silence | 3 |

Table 1: *Specification of the multi-levelled articulatory features used in this work. Cardinalities given in the right-hand column.*

We choose to work with OGI Numbers because of the detailed phonetic transcriptions which include diacritics where appropriate. A further consideration is that the data is useful for prototyping word recognizers due to the limited (30 word) vocabulary. The features, their values and cardinalities are listed in Table 1.

## 3. Artificial Neural Networks

ANNs were trained using the NICO Toolkit [9]. Recurrent time-delay neural networks consisting of three layers (input, hidden, output) were used, one for each feature group. Each network has one output unit for each value the feature can take. The numbers of hidden units used were: manner 300, place 300, voicing 100, rounding 200, front-back 250, and static 150. During training, input-output pairs consist of frames of acoustic parameters mapping to articulatory feature values. During testing, for each acoustic frame, the feature networks each output a value for each level that the feature can take. We interpret the set of outputs of a network as an (un-normalised) discrete posterior probability density function (PDF).

## 4. Dynamic Bayesian networks

A Bayesian network (BN) provides a means of encoding the dependencies between a set of random variables (RV), where the RVs and dependencies are represented as the nodes and edges of a directed acyclic graph. Missing edges (which imply conditional independence) are exploited in order to factor the joint distribution of all random variables into a set of simpler probability distributions. A dynamic Bayesian network (DBN) consists of instances of a Bayesian network repeated over time, with temporal dependency arcs linking the instances. The hidden Markov model (HMM) is a simple example of a DBN. The parameters of the model include conditional probability tables (CPTs) which describe the dependence of each discrete variable on its (discrete) parents' values, and conditional GMM distributions which describe continuous-valued variables dependence on their (discrete) parents' values.

### 4.1. AF recognition model topology

A set of inter-feature dependencies was derived for the task of articulatory feature recognition in [3]. The same model topology is used here, and is shown in Figure 3. In previous

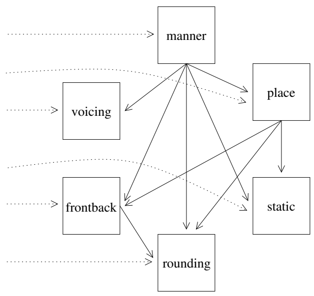


Figure 3: *Graph depicting inter-feature dependencies. Each feature is also conditioned on its value in the previous frame (implied by dotted arrows) and a silence/non-silence node which, along the observation node, has been omitted for clarity.*

studies [3, 4], the observation process comprised a product of Gaussian mixture models (GMM). The observation RVs were continuous-valued (MFCCs), and the GMMs were evaluated to provide the likelihood of each feature value generating a given observation vector. In this work, we use ANNs to provide virtual evidence (VE) [10], giving a model in the spirit of hybrid ANN/HMM ASR [11, 12]. The "observations" generated by the DBN are now the discrete-valued features themselves, but rather than observing the feature values directly, we incorporate

VE about them into the DBN in the form of a discrete PDF over each feature's possible values, with the PDFs provided by the ANNs. For a theoretical treatment of virtual evidence, see [10].

We incorporate virtual evidence as a scaled likelihood [11, 12]. The posterior $p_k(f_k|\,\mathbf{y}_t)$ associated with each level of feature $F_k$ is related to a generative likelihood by Bayes rule:

$$p_k(f_k|\,\mathbf{y}_t) = \frac{p_k(\mathbf{y}_t|\,f_k)\,p_k(f_k)}{p(\mathbf{y}_t)} \tag{1}$$

Ignoring $p_k(\mathbf{y}_t)$, which is independent of the feature state, the scaled likelihood is given as:

$$p_k(\mathbf{y}_t\,|\,f_k) \propto \frac{p_k(f_k\,|\,\mathbf{y}_t)}{p_k(f_k)} \tag{2}$$

# 5. Experiments

We first give a set of baseline results, then describe embedded training of feature CPTs and Viterbi training of the ANNs.

## 5.1. Baselines

A baseline for the hybrid ANN/DBN articulatory feature recognition presented in this paper is given by the hybrid ANN/HMM results reported in [13] (repeated in Table 2 below) for the same task. A hybrid ANN/HMM is a particular form of ANN/DBN in which feature streams are assumed independent of each other. Any accuracy improvement over the ANN/HMM results can be attributed to modelling of inter-feature dependencies.

| feature | ANN/HMM accuracy | ANN/DBN accuracy |
|---|---|---|
| manner | 84.6% | 88.9% |
| place | 81.6% | 87.2% |
| voicing | 84.2% | 87.2% |
| rounding | 84.7% | 88.5% |
| front-back | 84.1% | 88.2% |
| static | 81.6% | 86.6% |
| overall | 83.5% | 87.8% |

Table 2: *Test set results for ANNs (frame-level) and hybrid ANN/HMM system (segment-level). Column 2 taken from [13].*

ANNs were trained using phone-derived AF labels, and scaled likelihoods generated with priors calculated from training set feature-value frequencies. The only DBN parameters requiring training are the feature CPTs which assign probability to each value of a given feature conditioned on the values of its parents. These CPTs were estimated on the same set of time-aligned AF labels as used to train the networks. Feature insertion penalties were set using a held-out validation set. Test set results for hybrid ANN/DBN and ANN/HMM AF recognition are presented in Table 2. The performance is fairly uniform across the different features, with the ANN/HMM giving an overall accuracy of $83.5\%$. Modelling inter-feature dependencies in the ANN/DBN system gives an increase to $87.8\%$, amounting to a $26.1\%$ relative reduction in error.

## 5.2. Learning asynchronous feature changes

The conditional probability tables (CPTs) which describe the dependencies between features are sparse (i.e. most entries are zero). This dictates which features values can co-occur and which cannot. Training on phone-derived feature data leads to a very strong set of constraints on feature co-occurrence because only combinations which occur in the training data accumulate probability mass – the resulting *synchronous CPTs* are very sparse. The purpose of an AF approach is to model subtleties due to effects such as coarticulation and asynchronous movement of the production mechanism which are not compactly represented by phones. That is, we wish to learn which additional cells of the CPTs should have non-zero entries. In the absence of labels which give the level of detail required to train a set of asynchronous feature labels, we must infer this information directly from the acoustic data.

In earlier work [4], training *asynchronous CPTs* (less sparse than *synchronous CPTs*) for a feature recognition DBN, with MFCC observations and a GMM-based observation process, required a cascaded approach. The memory requirements for full inference with a 6-factorial hidden state were so great that asynchronous feature CPTs were trained for one feature at a time. However, in the current work, the scaled likelihoods used as virtual evidence are produced by ANNs (which are classifiers rather than generative models). Since ANNs are trained to discriminate between classes, they produce a larger spread in probability than GMMs (the discrete PDFs output by the ANNs have relatively low entropy). In combination with beam pruning, we find that full inference is now efficient enough to allow training of CPTs for all features simultaneously.

As in [4], we start with very sparse *synchronous CPTs* trained on phone-derived feature data, raise the zero-probability cells to some small value, renormalize and retrain with feature sequences (but not timings) given. The minimum value is $1/(\alpha\ \mathrm{card}(F_k))$, where $\mathrm{card}(F_k))$ denotes the cardinality of feature $F_k$, $\alpha$ is $10^5$, making the floor value an order of magnitude smaller than the smallest *synchronous CPT* cell value found after training on canonical labels.

| model | accuracy | # combinations |
|---|---|---|
| ANN/HMM | 83.5% | 2498 |
| ANN/DBN | 87.8% | 54 |
| ANN/DBN - asynch | 87.8% | 97 |

Table 3: *Summary of results for ANN/HMM and ANN/DBN AF recognition, the latter either with CPTs trained on phone-derived feature labels or where embedded training has been used to learn asynchronous changes. The number of feature combinations found in the decoded output is also given.*

Results, along with the number of feature combinations (i.e. the combination of 6 feature values in a single frame) found in the output are given in Table 3, for both canonically-trained (i.e. synchronous) and asynchronous feature CPTs. For comparison, the results using a hybrid ANN/HMM model are also shown.

We make two main observations from the results in this table: first, that decoding with the ANN/HMM model, in which feature streams are statistically independent, leads to substantially more feature combinations than the ANN/DBN model. We suspect many of these are spurious and result from small alignment errors between the independent feature stream.

The synchronous ANN/DBN gives a higher AF recognition accuracy (due to modelling of inter-feature dependencies), and more structure in the decoded output, i.e. fewer feature combinations. A number of these combinations correspond to a feature-encoding of phones. Embedded training of the feature CPTs (ANN/DBN - asynch), does not improve accuracy, though leads to an increase in the number of combinations found in the output. This suggests that a degree of asynchrony has been learned by reinforcing the likelihood of non phone-derive feature combinations. We expect to realize the benefits of this extra detail in a word recognition setting.

### 5.3. Viterbi training of the ANN observation process

Section 1.2 discussed the context of the current study, and stated our goal of using embedded training for automatic learning of pronunciation variation in terms of articulatory features.

The experiment presented in this section forms a precursor to this, and demonstrates that we are now able to perform Viterbi training of the ANNs. Our previous attempts have led to degeneration of the models. Viterbi training proceeds as follows: VE from ANNs trained using phone-derived feature labels is used in conjunction with the asynchronous-feature DBNs to realign the training set. The ANNs are then trained using the newly-aligned training labels. In the experiment reported below, we perform a single iteration of Viterbi training.

| feature labels | framewise validation accuracy | |
|---|---|---|
| | phone-derived | realigned |
| manner | 88.3% | 93.9% |
| place | 85.7% | 91.5% |
| voicing | 91.7% | 95.4% |
| rounding | 88.1% | 93.6% |
| front-back | 87.2% | 93.2% |
| static | 88.2% | 93.4% |
| overall | 88.2% | 93.5% |

Table 4: *Framewise validation set accuracies when ANNs are trained on phone-derived vs. realigned feature labels*

The framewise[1] classification accuracy on a held-out validation set is used to determine convergence during training. We find that for all features, these accuracies are higher after training on realigned feature labels than after training on the original phone-derived ones. The framewise accuracies at convergence are given in Table 4, and show an increase from 88.2% to 93.5% averaged over all features, suggesting that the realigned data leads to improved discrimination between feature values at the frame level. Articulatory feature recognition by the asynchronous ANN/DBN system yields a small, though not statistically significant increase in accuracy. Table 5 shows that Viterbi training leads an accuracy increase from 87.8% to 87.9%.

| model | accuracy |
|---|---|
| ANN/DBN | 87.8% |
| ANN (Viterbi) /DBN | 87.9% |

Table 5: *ANN/DBN articulatory feature recognition accuracy before and after a single iteration of ANN Viterbi training.*

The importance of this result is to show that we are able to perform Viterbi training without the models degenerating (as they had done in previous experiments). AF recognition is an important component of the feature-based word recognition system we are building, though has inherent limitations as a stand-alone task: models are trained and evaluated against feature sequences which carry the drawbacks of phones, as feature labels are derived from the phone labels. In particular, feature value insertions and deletions are not yet allowed, which limits the representation of co-articulation and assimilation effects. In the word-based system we are currently implementing, feature-based modelling of intra-speaker and pronunciation variation encodes a set of possible feature insertions and deletions. It is in this framework that we believe Viterbi training will yield true benefits.

[1] Other results reported here use ASR-style recognition accuracy.

## 6. Conclusions

In this paper, we have presented work which combines ANNs and DBNs for articulatory feature recognition. We have shown that by modelling the dependencies between feature streams we produce a 4.3% absolute, or 26.1% relative reduction in error. Furthermore, we have discussed how to refine the model through learning asynchronous changes where supported in the data, and shown feasibility of the Viterbi training which will be used in training our AF-based word recognition system.

**Acknowledgements:** thanks to Jeff Bilmes and Karen Livescu for answering GMTK-related questions. Also thanks to Mirjam Wester who provided the ANN training scripts.

## 7. References

[1] M. Ostendorf, “Moving beyond the ‘beads-on-a-string’ model of speech,” in *Proc. IEEE ASRU Workshop*, 1999.

[2] J. Bilmes, *GMTK: The Graphical Models Toolkit*, SSLI Laboratory, University of Washington, October 2002.

[3] J. Frankel, M. Wester, and S. King, “Articulatory feature recognition using dynamic Bayesian networks,” in *Proc of ICSLP-’04*, Jeju, Korea, 2004.

[4] M. Wester, J. Frankel, and S. King, “Asynchronous articulatory feature recognition using dynamic Bayesian networks,” in *Proc. IEICI Beyond HMM Workshop*, Kyoto, Dec. 2004.

[5] K. Kirchhoff, “Robust speech recognition using articulatory information,” Ph.D. dissertation, Berkeley, CA, 1998.

[6] S. King and P. Taylor, “Detection of phonological features in continuous speech using neural networks,” *Computer Speech and Language*, vol. 14, pp. 333–353, 2000.

[7] R. A. Cole, M. Noel, T. Lander, and T. Durham, “New telephone speech corpora at cslu,” in *Proc. Eurospeech*, vol. 1, Sep 1995, pp. 821–824.

[8] T. Lander, “The CSLU labeling guide,” Website, 15 May 1997, http://www.cslu.ogi.edu/corpora/docs/labeling.pdf.

[9] N. Ström, “Phoneme probability estimation with dynamic sparsely connected artificial neural networks,” *The Free Speech Journal*, vol. Issue #5, 1997.

[10] J. Bilmes, “On soft evidence in Bayesian networks,” University of Washington Department. of Electrical Engineering, Tech. Rep. UWEETR-2004-0016, 2004.

[11] N. Morgan and H. Bourlard, “Neural networks for statistical recognition of continuous speech,” *Proceedings of the IEEE*, vol. 83, no. 5, pp. 741–770, May 1995.

[12] A. Robinson, G. Cook, D. Ellis, E. Fosler-Lussier, S. Renals, and D. Williams, “Connectionist speech recognition of broadcast news,” *Speech Communication*, vol. 37, pp. 27–45, 2002.

[13] J. Frankel, M. Wester, and S. King, “Articulatory feature recognition using dynamic Bayesian networks,” *In preparation*, 2005.