Articulatory-acoustic Feature Recognition: Comparison of Machine Learning and HMM methods

Jan Macek, Supphanat Kanokphara, Anja Geumann

Department of Computer Science
University College Dublin
{jan.macek, supphanat.kanokphara, anja.geumann}@ucd.ie

Abstract

HMMs are the dominating technique used in speech recognition today since they perform well in overall phone recognition. In this paper, we show the comparison of HMM methods and machine learning techniques, such as neural networks, decision trees and ensemble classifiers with boosting and bagging in the task of articulatory-acoustic feature classification. The experimental results show that HMM methods work well for the classification of such features as vocalic. However, decision tree and bagging outperform HMMs for the fricative classification task since the data skewness is much higher than for the feature vocalic classification task. This demonstrates that HMMs do not perform as well as decision trees and bagging in highly skewed data settings.

1. Introduction

Articulatory-acoustic Features (AF, also called articulatory features or phonological features by other authors) have been shown to improve word recognition accuracy under variable conditions of speech signal production. In the multilingual setting, feature recognizers trained on data from different languages were shown to have the capability of improving the overall performance by use of an ensemble or cross-lingual recognizer [1, 2]. These properties of AF recognizers motivate our research; in what follows we concentrate on AF extraction in a single language setting.

AFs are thought to be a good compromise between a better description of the acoustic signal (than phonemes) and still providing a linguistically interpretable symbolic annotation for a further processing system as in [3]. Acoustic correlates of features have been described extensively in the literature [4, 5]. The first detailed description of distinctive features [6] assumed that they had identifiable counterparts.

While HMMs have been predominantly used as acoustic models in current speech recognition systems, they have not been used frequently for AF extraction. A number of machine learning techniques as multilayer perceptron, dynamic Bayesian networks, support vector machines, and recurrent neural networks have been suggested for use in AF based speech recognition tasks [7, 8, 9, 10]. We focus here on a comparison of techniques for AF recognition of manner features.

As mentioned above one of the biggest potential advantages of feature-based speech recognition is its cross-lingual usability. This, however, requires a definition of language independent feature sets.

Based on these concerns [11] a detection of individual AFs, e.g. voc+ vs. voc-, nas+ vs. nas- as opposed to a more tier-based detection of separate values as {vocalic, stop, fricative, nasal, etc.} for a manner feature [7, 8] is favoured. In a multilingual framework a wide range of feature combinations has to be allowed, e.g. French nasal vowels will not be identified correctly if only one manner feature can be detected. The detection of individual features will often present highly skewed data distributions.

The material we use here (TIMIT) is phonetically fairly balanced; however this might not always be the case. Phonetically unbalanced material can be another source of data skewness, although of course even phonetically balanced material will show an unequal frequency distribution between sounds and features. It is this latter type of skewness which we consider here.

In the following we will use two manner features which seem to be fairly robustly recognized, *vocalic* and *fricative* to explore different pattern classification (machine learning and HMM) techniques. *Fricative* is a feature which has a considerable higher skewness than *vocalic* as fricatives will in general occur less frequently than vowels. Differences in recognition results between the two could then be interpreted as being related to skewness of the data. It was suggested [12] that certain machine learning techniques might cope better with highly skewed class distributions.

2. Methods and Experimental Setup

2.1. TIMIT corpus

The TIMIT corpus [13] of American English consists of 6300 sentences, 10 sentences spoken by 630 speakers. For training we used the whole training set (4620 utterances) and for testing the core test set (192 utterances). The manually annotated data were labeled with the corresponding features similar to [7]. The mapping of features to sound classes is described in Table 1 for *vocalic* and *fricative*.

Table 1: Sound-feature correspondence for *vocalic* and *fricative*

Feature	Arpabet sounds
fric+	f v th dh s z sh zh ch jh hh hv
fric-	ae aa aw ay ah ao ax axr ax-h ey eh er iy ih ix oy ow uw uh ux p b t d k g dx q m n ng em en eng nx l el r w y
voc+	aa ae ah ao aw ax ax-h ay eh er ey ih ix iy ow oy uh uw ux
voc-	p b t d k g dx q m n ng em en eng nx l el r w y axr f v th dh s z sh zh ch jh hh hv
sil	pau h# epi

2.2. Machine learning techniques

The machine learning techniques used have been described in more detail in [14]. The methods described in the following were selected as examples representative of different approaches to symbolic machine learning. The discrete function estimation based on decision trees generates an attribute-based model and performs logical tests on these attributes to classify the data.

In contrast to hidden Markov model based systems, the machine learning approaches presented here classify each frame independently while HMM-based systems search for a contextually optimized sequence of all frames in an utterance.

As a description of the speech signal we used 10ms frames that were analysed for attributes used for further learning and classification.

These attributes used for learning and classification were: energy of the current frame, energy of the 5 preceding frames, entropy of the current frame, entropy of the 5 preceding frames, energies in four frequency bands of the current frame (0-1.5 kHz, 1.5-3 kHz, 3-6 kHz, 6-8 kHz).

2.2.1. Decision trees with C4.5 as an example of an estimator based on discrete functions

The decision tree represents knowledge gained during the learning phase in the form of nodes and leaves. Each node bears a test on attribute values that are used for description of the data. Leaves of the tree represent the corresponding classes for conjunction of logical tests on the way from the root of the tree to the leaf. This formalism allows users to easily understand the encoded knowledge and to transform it to a set of logical rules, if desired.

Efficient methods for decision tree construction exist, e.g. C4.5 [15], CART, OC1. In our experiments, we used the C4.5 algorithm with Reduced Error Pruning, a technique used for simplification of the generated trees.

2.2.2. Neural Networks as an example of an estimator based on continuous functions

The neural network approximates the distribution generating the training data in the form of a network of simple threshold units and weighted connections and sums of outputs of these simpler units. The threshold units are typically nonlinear functions of the input variable. Neural networks are capable of learning complicated nonlinear functions and the setting of the topology plays a number of crucial roles in managing the trade-off between the speed of learning of the network and the achievable accuracy.

In our experiments we used neural networks with simple perceptron elements.

2.2.3. Ensemble classifiers with Boosting (AdaBoost.M1)

The main motivation behind Boosting is to combine weaker/simpler classifiers in an ensemble in a way that improves the performance of the combined classifier. Thus the performance of a single ensemble element is improved - i.e. boosted. Let hypotheses $h_1, \ldots, h_m$ form the set of hypotheses used in an ensemble, that forms a combined hypothesis as presented in Equation 1. The hypotheses h_i and coefficients α_i of the ensemble are learned with the boosting procedure, that is a hypothesis is learned iteratively on set of weighted examples and its weight is set according to its accuracy on the set of examples. According to the accuracy of the hypothesis the weights of the instances in the training set for the next iteration are determined.

2.2.4. Bagging with REP pruned decision trees

Bagging was introduced by Breiman (see [16]). It is a technique that uses bootstrap samples for the construction of replicate classifiers and a combination technique that weights outputs of individual classifiers to yield final classification.

The bootstrapped sample is created by drawing with replacement n examples from training set $S_n = (x_i, y_i), i=1,\ldots,n$. With the same size as the original training data, it contains replicates of some examples, while others are not presented. Typically, bootstrap sampling is performed multiple times (25–50 times). Training on each of the training sample is performed by traditional machine learning technique and the bagged estimate is obtained by averaging the resulting estimator that we describe as

$$f(x) = \sum_{t=1}^{m} \alpha_t h_t(x) \qquad (1)$$

where h_i is a base classifier trained on bootstrap sample i, α_i is the averaging constant and $f(x)$ is the resulting ensemble classifier. As the base classifiers we used the decision trees with REP described in section 2.2.1.

2.3. The HMM-Based AF Extraction System

The HMM, by design, is used to map some uncertainty signal into a sequence of units. These units can be words, syllables, demi-syllables, phones, etc. The articulatory feature extraction presented here also uses this type of HMMs to map a speech signal into a sequence of features.

2.3.1. System overview

The HMM-based articulatory feature extraction system in this paper is constructed using HTK [17]. The acoustic model training system starts by converting the speech signal into a sequence of vector parameters with a fixed 25 ms frame and a frame rate of 10 ms. Each parameter is then pre-emphasized with the filter *P(z) = 1-0.9*z^(-1)*. The discontinuities at the frame edges are attenuated by using Hamming window. A fast Fourier transform is used to convert time domain frames into frequency domain spectra. These spectra are averaged into 24 triangular bins arranged at equal mel-frequency intervals (where $f_{mel} = 2595 \log_{10}(1+f/700))$, where f denotes frequency in Hz. 12 dimensional mel-frequency cepstral coefficients (MFCCs) are then obtained from cosine transformation and lifter. The normalized log energy is also added as the 13th front-end parameter. The actual acoustic energy in each frame is calculated and the maximum selected. All log energies are then normalized with respect to maximum and log energies below a silence floor (set to -50 dB) clamped to that floor.

These 13 front-end parameters are expanded to 39 front-end parameters by appending first and second order differences of the static coefficients. The chosen parameters chosen have been used extensively and have proven to be one of the best choices for HMM-based speech recognition systems.

Each model contains 5 states with no skip state and the covariance matrices of all states are diagonal. After the context-independent HMMs have been trained, they are expanded to context-dependent HMMs by using cross-feature network, backing-off technique [18]. Maximum likelihood estimators are used to train HMM parameters. The number of training iterations after each change is determined automatically. The model mixtures are expanded one by one until the model is saturated. For the recognition process, the Viterbi algorithm is used without any pruning factor and language model.

2.4. Experimental Results

In the experiments we estimated the accuracy of classifiers by their performance on isolated frames which is motivated by the nature of machine learning methods as opposed to time-mediated or string-alignments.

Table 4 presents the data skewness for both features in the training data. We present accuracies of the classifiers in Table 5 for the classification of the feature *vocalic* and for the feature *fricative*. Based on the accuracies in Table 5 for *fricative,* ensemble classifiers with bagging perform best, followed by decision trees (C4.5). For *vocalic* HMMs perform best followed by ensemble classifiers with bagging.

Table 4: Data skewness: Relative distribution for *fricative* and *vocalic* in training material

fric+	fric-	sil
16.1%	75.9%	8.1%
voc+	voc-	sil
37.9%	55.7%	8.1%

Table 5: The accuracy of classification for features *fricative* and *vocalic*

Classifier	fricatives	vocalic
C4.5 (decision tree)	86.4 %	76.4 %
Neural Network	82.3 %	70.0 %
Boosting (AdaBoost.M1)	75.6 %	66.5 %
Bagging (with REPTrees)	**88.0 %**	77.9 %
HMM	83.1 %	**81.4 %**

To understand the differences between achieved accuracies better under highly skewed data conditions, we require the following measures. The precision is defined as the ratio

$$\frac{\#\,\text{of correctly classified instances of class}\ c}{\#\,\text{of instances classified as class}\ c} \qquad (2)$$

and the recall is defined as the ratio

$$\frac{\#\,\text{of correctly classified instances of class}\ c}{\#\,\text{of instances of class}\ c} \qquad (3)$$

Both measures analyse the correct classification for each feature class individually. It is important to mention the trade-off between precision and recall. The aim is to achieve high values for both while each of them act against each other. This trade-off is usually described by the *F-Measure* that is defined as the ratio

$$\frac{2 \times precision \times recall}{precision + recall} \qquad (4)$$

A more in depth look at *precision* and *recall* in Tables 6 and 7 make differences between the different techniques much more obvious. The boosting method seems to fail in the case of *fric+* and *sil*. Both classes are entirely misclassified as *fric-*. In the case of boosting for vocalic only *sil* in Table 7 is entirely misclassified; in the less skewed data *voc+* and *voc-* the boosting does perform better.. We conclude that the skewness of the data might effect boosting more than the other methods.

While the decision tree and bagging perform similarly to HMM in terms of the precision and recall, and the F-Measure (see Tables 6 and 7), the overall accuracy favours bagging and decision trees. As in the case of poor performance of boosting, we believe that the high skewness of the data in the case of the feature *fricative* is the source of better performance of bagging and decision trees. The F-measure values for HMM (see Table 7) make it apparent that HMMs tend to produce classifiers with more uniform class distributions which results in its poorer performance on the highly skewed class of fricatives.

In the case of feature *vocalic* classification the performance of HMM is clearly best using all presented measures.

Table 6: The F-measure for corresponding classifiers for features *fricative* and *vocalic*

	C4.5	Neural Network	Bagging with REPTrees	Boosting (AdaBoost.M1)	HMM
fric-	0.921	0.895	**0.930**	0.861	0.909
fric+	0.649	0.484	**0.682**	0.000	**0.684**
sil	0.708	0.636	0.760	0.000	**0.880**
voc-	0.783	0.689	0.798	0.657	**0.867**
voc+	0.747	0.721	0.757	0.731	**0.855**
sil	0.708	0.633	0.753	0.000	**0.808**

3. Discussion and Conclusions

We have demonstrated that AF recognition can be performed fairly well with HMMs and other machine learning techniques such as bagging or decision trees.

Comparing the results of HMM and best of the presented machine learning techniques we could conclude that HMM performance is better on less skewed data as for *vocalic*. The machine learning technique of decision trees outperformed HMM on more skewed data, i.e. *fricatives* as can be seen in Table 5.

As we can see, the accuracies of machine learning methods in Table 5 for the feature *fricative* are even better than for *vocalic*. Data skewness does not seem to affect the performance to similar extent to the case of fricatives since the data for the feature *fricative* are more skewed than for *vocalic*.

Table 7: The precision and recall rates for corresponding classes and classifiers for features *fricative* and *vocalic*

	C4.5		Neural Network		Bagging with REPTrees		Boosting (AdaBoost.M1)		HMM	
Feature class	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall
fric-	0.901	0.941	0.850	0.944	0.910	0.951	0.756	1	0.850	0.976
fric+	0.734	0.581	0.676	0.377	0.771	0.611	0	0	0.906	0.550
sil	0.710	0.706	0.690	0.590	0.764	0.757	0	0	0.897	0.863
voc-	0.794	0.773	0.831	0.588	0.805	0.791	0.758	0.579	0.859	0.875
voc+	0.729	0.766	0.601	0.901	0.739	0.775	0.597	0.943	0.942	0.783
sil	0.723	0.693	0.756	0.544	0.797	0.714	0	0	0.762	0.861

We suspected the inherent string alignment of HMMs to be less helpful for AF extraction. Further we expected that the documented usefulness [12] of machine learning techniques in skewed data conditions would be a greater advantage.

It would be interesting to take the better of both approaches and enhance the machine learning with contextual information that is one of the distinguishing properties of HMMs. Combination of classifiers could further improve the performance on the presented task.

Future work is concerned with a more detailed error analysis providing further information which may result changing the sound-feature specification. We suspect that the feature *voc+* could be used for vowels and approximants. The feature *fric+* might better be identified as *frication* and then include stop bursts.

Further improvements can also be expected by adaptation of the attributes to psychophysical salient characteristics in a similar direction to [2, 6].

4. Acknowledgements

This material is based upon works supported by the Science Foundation Ireland under grant No. 02/INI/II00. The opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of SFI.

5. References

[1] S. Stueker, F. Metze, T. Schulz, and A. Waibel. Integrating Multilingual Articulatory Features into Speech Recognition. In *Proceedings of EUROSPEECH* 2003, pages 1033–1036, 2003.

[2] S. Stueker, T. Schulz, F. Metze, and A. Waibel. Multilingual Articulatory Features. In *Proceedings of ICASSP* 2003, vol. 1, pages 144–147, 2003.

[3] Carson-Berndsen, J. *Time Map Phonology: Finite State Models and Event Logics in Speech Recognition.* Dordrecht, Holland: Kluwer Academic Publishers, 1998.

[4] Stevens, K.N. Acoustic correlates of some phonetic categories, *JASA* 68(3):836-842, 1980.

[5] Stevens, K.N. *Acoustic Phonetics*, MIT Press: Cambridge (Ma), London, 1998.

[6] Jakobson, R., Fant, G., Halle, M. *Preliminaries to speech analysis: The distinctive features and their correlates*. MIT Press, 9th ed. 1969 (1952).

[7] Chang, S., Greenberg, S. and Wester, M. An Elitist approach to articulatory-acoustic feature classification. In *Proc. 7th Eurospeech*, Aalborg, Denmark, pages 1725-1728, 2001.

[8] Frankel, J. Wester, M. & King, S. "Articulatory Feature Recognition Using Dynamic Bayesian Networks", *Proc. Intl. Conf. on Spoken Language Processing*, Jeju, Korea, pages 1202-1205, 2004.

[9] Hasegawa-Johnson, M., Baker, J., Borys, S., Chen, K., Coogan, E., Greenberg, S., Juneja, A., Kirchhoff, K., Livescu, K., Mohan, S., Muller, J., Sonmez, K. & Wang, T. "Landmark-based Speech recognition: Report of the 2004 Johns Hopkins Summer Workshop". In *Proc. of ICASSP* 2005.

[10] K. Hacioglu, B. Pellom, and W. Ward. Parsing Speech into Articulatory Events. In *Proceedings of ICASSP* 2004, volume 1, pages 925–928, 2004.

[11] Geumann, A. Towards a new level of annotation detail of multilingual speech corpora. In *Proc. Intl. Conf. on Spoken Language Processing*, Jeju, Korea, pages 1096-1099, 2004.

[12] Liu, Y., Shriberg, E., Stolcke, A. & Harper, M. "Using Machine Learning to Cope with Imbalanced Classes in Natural Speech: Evidence from Sentence Boundary and Disfluency Detection", *Proc. Intl. Conf. on Spoken Language Processing*, Jeju, Korea, pages 660-663, 2004.

[13] Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S. & Dahlgren, N.L., *DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus* CDROM, NIST, 1993.

[14] Witten, I.H. & Frank, E., *Data Mining: Practical machine learning tools with Java implementations*, Morgan Kaufmann: San Francisco, 2000.

[15] Quinlan, R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[16] Breiman, L. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[17] Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V. and Woodland, P. 2002. *The HTK Book*, Microsoft Corporation and Cambridge University Engineering Department, December.

[18] Kanokphara, S. and Carson-Berndsen, J. 2005. Better HMM-Based Articulatory Feature Extraction with Context-Dependent Model, In *Proc. of the 18th International Florida Artificial Intelligence Research Society Conference.*