Articulatory Feature Classification Using Nearest Neighbors

Arild Brandrud Næss[1], *Karen Livescu*[2], *Rohit Prabhavalkar*[3]

[1]Department of Electronics and Telecommunications,
Norwegian University of Science and Technology, Trondheim, Norway
[2]Toyota Technological Institute at Chicago, Chicago IL, USA
[3]Department of Computer Science and Engineering,
The Ohio State University, Columbus OH, USA

arild.naess@iet.ntnu.no, klivescu@ttic.edu, prabhava@cse.ohio-state.edu

Abstract

Recognizing aspects of articulation from audio recordings of speech is an important problem, either as an end in itself or as part of an articulatory approach to automatic speech recognition. In this paper we study the frame-level classification of a set of articulatory features (AFs) inspired by the vocal tract variables of articulatory phonology. We compare k nearest neighbor (k-NN) classifiers and multilayer perceptrons (MLPs), using different acoustic feature vectors, and classify the AFs either independently or jointly. We also consider using the MLP outputs for all of the AFs as inputs to k-NN classifiers for the individual AFs, effectively using the MLPs as a form of nonlinear dimensionality reduction and allowing the decision for each AF to be based on the MLPs for the other AFs. We find that MLPs outperform k-NN classifiers, while k-NN classifiers using MLP outputs outperform both.

Index Terms: articulatory modeling, speech inversion, multilayer perceptron, k nearest neighbors

1. Introduction

In this paper we address the problem of classifying articulatory features (AFs) from audio. This task has been addressed for many reasons, including on its own as a scientific question and as an intermediate step in articulatory approaches to automatic speech recognition. In this work we consider a particular set of multi-valued AFs based on the vocal tract variables of articulatory phonology [1], representing the locations and constriction degrees of the lips, tongue tip, and tongue body and the states of the velum and glottis. These features, shown in Table 1, have been proposed as alternative sub-word units (as opposed to phones) for automatic speech recognition [2, 3, 4, 5].

Compared to other AF sets, such as those based on phone categories in the International Phonetic Alphabet (IPA), the feature set we consider has certain appealing qualities, such as greater independence of the features and a more direct correspondence with measurable vocal tract properties. In addition, in lexical access experiments, where a word is identified from its AF values, this feature set was more successful than one based on IPA categories [4].

The classification of articulatory phonology-based features from acoustics has not been extensively studied. In this paper, we study the problem of classifying the feature values in a frame of speech, both jointly and independently, using k nearest neighbor (k-NN) classifiers and multilayer perceptrons (MLPs). MLPs are commonly used for phonetic classification, and have also been used for classification of IPA-style AFs [6, 7], but to our knowledge not for articulatory phonology-based features. The outputs of MLPs are often used in speech recognizers indirectly, by learning Gaussian mixture models over their outputs in so-called tandem systems [8].

While nearest neighbor-based methods are not often used in speech recognition, there has been some recent effort in this direction (e.g., [9]). nearest neighbor classifiers involve no training; the training data is simply stored, and test points are classified by finding the nearest neighbors to it among the training points. For example, Deselaers *et al.* [9] use the k nearest neighbors of a given test frame to build kernel density estimates of state likelihoods in a hidden Markov model-based speech recognizer. k-NN classifiers have some advantages that make them particularly attractive for speech recognition: they are inherently discriminative, they avoid the long training times associated with typical Gaussian mixture models on large data sets, and they are nonparametric so they involve relatively little tuning (typically, only the number of neighbors k). The nonparametric nature of k-NN classifiers is particularly attractive for new types of models, such as articulatory models, where there is less understanding of the distribution of the data or of the best strategy for training a parametric classifier.

Nearest neighbor techniques have one major disadvantage: at test time, they require a search through all of the training data. However, a number of approximate search techniques can greatly improve search speed (e.g., [10]).

We also consider ways of taking into account the relationships between the multiple AF classification tasks. While the AFs are "physiologically" largely independent (there are few constraints on the state of one articulator given those of other articulators), it may be helpful to consider the tasks together. To this end, we classify the features both jointly and independently. We also consider a "tandem-like" approach in which the outputs of the MLPs are used as inputs to k-NN classifiers. This has the advantage of implicitly taking into account the correlations between the multiple AF classification tasks: By using the MLP outputs for all of the AFs, each AF classifier benefits from information about the MLPs' outputs for the other AFs.

2. Methods

2.1. k Nearest Neighbors

The k-NN algorithm consists of classifying each example $\mathbf{x}$ in the test set by majority vote of the k nearest examples from the

This work was supported by the SIRKUS project under the Research Council of Norway's VERDIKT programme, and by NSF grants IIS-0905420 and IIS-0905633.

Table 1: Our articulatory feature set.

feature	values
LIP-LOC	protruded (0), labial (1), dental (2)
LIP-OPEN	closed (0), critical (1), narrow (2), wide (3)
TT-LOC	inter-dental (0), alveolar (1), palato-alveolar (2), retroflex (3)
TT-OPEN	closed (0), critical (1), narrow (2), mid-narrow (3), mid (4), wide (5)
TB-LOC	palatal (0), velar (1), uvular (2), pharyngeal (3)
TB-OPEN	closed (0), critical (1), narrow (2), mid-narrow (3), mid (4), wide (5)
VEL	closed (0), open (1)
GLOT	closed (0), critical (1), wide (2)

training set, where "nearest" is defined through a metric in the input feature (observation) space. [1] Given a feature space $\mathcal{X}$, a label space $\mathcal{Y}$, a training set $\mathcal{T} = \{(x_i \in \mathcal{X}, y_i \in \mathcal{Y})\}_{i=1}^{N}$, a metric $\mathcal{D} : (\mathcal{X} \times \mathcal{X}) \mapsto \mathbb{R}$, a natural number $k < N$ and a test example $x_0 \in \mathcal{X}$; let $i_1, ..., i_k$ be the indices of the k nearest neighbors of x_0 in $\mathcal{T}$ with respect to $\mathcal{D}$, and for each $y \in \mathcal{Y}$, let X_y be the subset of those nearest neighbors belonging to class y, i.e., $X_y = \{i_j : y_{i_j} = y, 1 \leq j \leq k\}$. Then, the class of x_0 is predicted by majority vote: $\hat{y}_0 = \arg\max_{y \in \mathcal{Y}} |X_y|$. Ties can be resolved by considering a lower value of k. The case $k = 1$ corresponds to the basic nearest neighbor algorithm.

The basic assumption behind nearest neighbor classification schemes is that examples that are close in the feature space are likely to belong to the same class. How well this works depends on the distance measure $\mathcal{D}$. Typical choices are the Euclidean and Mahalanobis distances. The Mahalanobis distance is given by $\mathcal{D}(x_1, x_2) = \sqrt{(x_1 - x_2)^T M (x_1 - x_2)}$, where M is the Mahalanobis matrix. If M equals the identity matrix, this reduces to the Euclidean distance. Another common choice for M is the inverse covariance of the data, which is equivalent to whitening the data and then using the Euclidean distance.

Linear feature transforms, such as principal components analysis (PCA) and linear discriminant analysis (LDA), also induce Mahalanobis distances. Consider linear transform L from $\mathbb{R}^d$ to another (possibly lower-dimensional) space $\mathbb{R}^p$. The Euclidean distance between two vectors mapped to this space, $\mathcal{D}(Lx_1, Lx_2)$, is equivalent to a Mahalanobis distance with $M = L^T L$ in the original space:

$$(L(x_1 - x_2))^T I L(x_1 - x_2) = (x_1 - x_2)^T L^T L(x_1 - x_2).$$

Estimating such a transform is therefore equivalent to learning a Mahalanobis distance measure from the data.

There are many extensions to this approach, including various distance learning techniques (e.g., [11, 12]) and extensions of the k-NN rule that include distance-dependent weighting of the neighbors. In this work, we concentrate on the classic k-NN rule with Euclidean distances or the Mahalanobis distances induced by PCA and LDA.

2.2. Multilayer Perceptrons

Multilayer perceptrons (MLPs) have been widely used in speech recognition as part of models based on hidden Markov models (HMMs). Typically, MLP phone (or phone state) classifiers are used in either a so-called "hybrid approach" [13], in which HMM emission probabilities are replaced by scaled likelihoods computed from MLP phone posteriors; or in a "tandem approach" [8], in which the MLP log-posteriors serve as a replacement for (or an addition to) the raw acoustic features in a conventional GMM-HMM system. The tandem approach uses MLPs as a form of discriminative, nonlinear dimensionality reduction. MLP-based articulatory feature classifiers have also been used in these frameworks, though with very different types of articulatory features than the ones we consider here [6, 7].

Given a training set $\mathcal{T} = \{(x_i \in \mathcal{X}, y_i \in \mathcal{Y})\}_{i=1}^{N}$, denote by $p_i(y)$ the empirical distribution of the posteriors over the classes y, given an input example x_i. Thus, $p_i(y) = 1$ if and only if $y = y_i$, and is 0 otherwise. In our experiments, we use MLPs with one hidden layer, with a softmax non-linearity on the output layer nodes. The outputs $q_i(y)$ can be interpreted as an estimate of the posterior distribution over the classes y given the input vector x_i. Training consists of adjusting the weights $\mathbf{w}$ by error backpropagation to minimize the following relative entropy (K-L divergence) criterion:

$$\mathcal{L}(\mathbf{w}) = \sum_{i=1}^{N} \sum_{y \in \mathcal{Y}} p_i(y) \frac{\log p_i(y)}{\log q_i(y)}$$

2.3. Using MLP Posteriors for k-NN Classification

Motivated by the success of tandem approaches in speech recognition, we also consider an approach that combines MLP outputs with k-NN classification. As an alternative to the k-NN metrics based on linear transformations, such as PCA and LDA, we consider the nonlinear mapping given by class posteriors from MLPs trained to classify AFs. In contrast to other work using phone-based MLPs, we are addressing multiple classification tasks and will have multiple MLPs. For each articulatory feature, we have a choice of either using only the MLP posteriors (MLPPs) for that feature or concatenating posteriors from multiple MLPs. We hypothesize that using concatenated MLPPs to classify each AF may be preferred, as it allows us to implicitly take into account the dependencies between AFs. Because of these dependencies, it may be helpful to know, when classifying one AF, the "opinions" of MLPs for other AFs.

3. Experiments

Our experiments are done on a portion of the Switchboard Transcription Project (STP) data [14]. The STP data is a subset of the Switchboard conversational corpus that was manually transcribed at a detailed phonetic level, including diacritics indicating nasalization, frication (of an otherwise unfricated sound), and so on. This allows us to consider the labeling as a proxy for articulatory feature labels. We use only the portion of STP that was labeled and aligned one segment at a time (the remainder of STP was labeled at a syllabic level). We choose this corpus because of its detailed labeling and conversational nature; we expect the biggest advantages of using articulatory models over phonetic models to be seen for conversational, spontaneous speech rather than more formal/planned speech.

The ground-truth values for each of the AFs are set by a deterministic mapping from the annotated phones. This mapping is laid out in detail in [4]. Stops, affricates, and diphthongs are split into two segments each with separate AF configurations. In such cases, 2/3 of the annotated segment's duration is assigned to the first configuration and the latter 1/3 to the second. Mapping from phones to AFs is not optimal, but we choose this option due to the lack of corpora annotated at the articulatory feature level. In this work we are considering classification on the frame level, so we break up each segment into 10ms frames.

[1] Note that we are using two senses of the word "feature": The input *acoustic features* are the observations measured from the speech signal; and the classifier outputs are the discrete *articulatory features*.

The eight AFs listed in table 1 can be classified separately; or they can be classified jointly, treating each configuration of all eight AFs as one label. The former approach is more appropriate under the assumption that there are no dependencies between the AFs, while the latter allows for arbitrary dependencies (intermediate options exist as well; for now we consider the two extremes). For the former approach, the number of classes for each classification task is the cardinality of the AF in question. In the latter case, although the joint state space of the AFs has size 41 472, only 66 configurations occur in the data and we limit ourselves to this number of classes.

The acoustic observations are standard 13-dimensional mel-frequency cepstral coefficients (MFCCs) plus their first and second derivatives, computed for each 10ms frame. We also experiment with concatenations of multiple consecutive frames around the current frame to form longer-context feature vectors. Here we report on experiments using either single-frame windows or 9-frame windows (resulting in 211-dimensional feature vectors). For the 9-frame windows, we experiment with dimensionality reduction using PCA and LDA. The frames are assigned ground-truth phonetic labels based on the time stamps in the transcriptions. Silence frames are excluded, so that silence does not dominate the classification task by virtue of being by far the most frequent label. After excluding silence frames, our corpus consists of 219 251 frames, or 36.5 minutes, of speech.

All of our experiments are done on the training subset of the STP transcriptions. We use subsets 20, 23–49 for training and tuning and subsets 21–22 as test data. For the MLP experiments and for the k-NN experiments using MLP posteriors, we tune on subset 20. For the other k-NN experiments, the hyperparameters (k, the dimensionality for PCA and LDA) are tuned by five-fold cross-validation over subsets 20 and 23–49. Training (when applicable) with the optimal parameters is done over all of sets 20, 23–49, before testing on subsets 21–22. We use freely available Matlab toolboxes for the MFCC computation, dimensionality reduction, and k-NN classification [15, 16, 11]. We trained the MLP classifiers using the QuickNet toolkit [17].

The MLPs have a single layer of hidden units with a sigmoid activation function and a softmax activation function on the output layer units as described in Section 2.3. The number of units in the hidden layer was tuned on the development set. To avoid overfitting, we adopted an adaptive learning rate schedule and early stopping based on classification performance on the held-out development set.

For k-NN classification using MLP posteriors (MLPPs), we consider several setups: using only the MLPPs corresponding to a single AF for that AF's classification; using the concatenation of all of the MLPPs for each AF; and concatenating either of these with the MFCC vectors (either single-frame and 9-frame windows) to create a "tandem-like" feature vector. We also consider using either raw posterior probabilities or log-posteriors in these setups. Of these, we do not give results for MFCC + MLPP vectors or for log-posteriors, because in preliminary experiments they consistently did worse than the other options.

In order to give an idea of how the methods perform on a more familiar task, we also repeat some of our experiments on the task of phonetic classification on the same data.

4. Results and Discussion

Our results for the classification of phones and AFs are summarized in Figure 1 and Table 2. For phone classification, the 9-frame MLP outperforms the 9-frame k-NN classifier by about 19% relative. The MLPP k-NN, however, improves further on the MLP, attaining an error rate of 60.07%. This compares favorably to a previously reported error rate of 62.7% for a 9-frame MLP on a similar (but not identical) subset of STP [18].

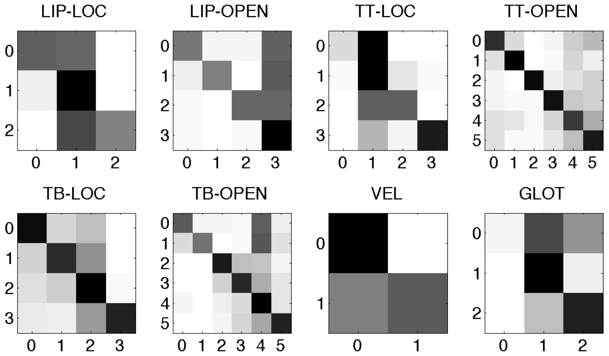

Figure 2: Confusion matrices for k-NN AF classification using MLPP inputs. Rows corresponds to true classes and columns to predicted classes. Darker cells indicate higher values.

For AF classification, we see the same pattern: The MLPs do better than basic k-NN classifiers by a large margin, and MLPP k-NN outperforms the MLPs for all eight classification tasks, with relative improvements ranging from 2% to 6.4%. While these improvements are small, they are encouraging because they suggest that, when we integrate the classifiers into a speech recognition system, it may not be necessary to do further training using the MLP outputs (as in tandem models).

Most trends in the results are consistent across the eight AF classification tasks. Concatenating features across nine frames substantially improves both k-NN classifiers and MLPs. Reducing the dimensionality of the concatenated features for each AF by PCA or LDA before k-NN classification gives little or no improvement. We hypothesize that, in the case of LDA, too few dimensions are retained (LDA is limited to one dimension fewer than the number of classes). In the case of PCA (where our tuned dimensionalities range from 25 to 125), we hypothesize that Euclidean distance in the transformed space may be a less appropriate measure than in the raw observation space.

In addition to the standard PCA and LDA, we also consider learning an LDA matrix for the joint AF classification task and using it for each of the individual classification tasks. This substantially improves over the other linear dimensionality reduction techniques and over using the full-dimensionality vectors.

Using the MLP posteriors as a nonlinear dimensionality reduction proves to be the most successful. From our different setups for k-NN classification with MLP posteriors, we get the best results using raw rather than log probabilities, and using the MLPP features alone rather than concatenated with MFCCs. We do, however, see a significant improvement from combining the posteriors from all eight MLPs for each k-NN classification.

The numbers of neighbors, k, that give the best results for the MLPP k-NN classifiers (as measured by cross-validation performance) are around 200 for all AFs except VEL, for which it is 31. It should be noted, however, that the performance is not very sensitive to the choice of k above about 50. Thus, when computational complexity is an issue, lower values of k can be used without substantial loss in performance.

Joint k-NN classification performs worse than individual classification for all AFs. Using MLPPs improves the results, but they are still worse than the individual classifiers with full dimensionality.

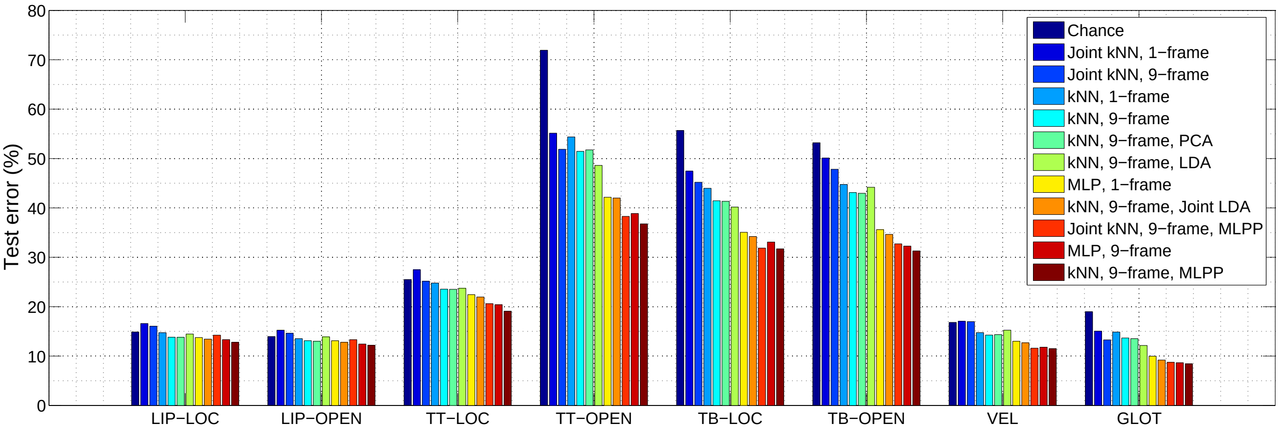

Figure 1: Error rates for AF classification. Methods sorted in order of mean performance. “Chance” refers to assigning each test vector the most frequent class label in the training data.

Table 2: Error rates for the phone and articulatory feature classifications, in percent. “Raw” indicates no dimensionality reduction.

	Chance	*k* **Nearest Neighbors**									**MLP**	
		Joint classification			**Individual classification**							
		1-frame	*9-frame*		*1-frame*	*9-frame*					*1-frame*	*9-frame*
		raw	raw	MLPP	raw	raw	PCA	LDA	joint LDA	MLPP		
LIP-LOC	14.9	16.6	16.1	14.2	14.7	13.8	13.8	14.5	13.4	**12.8**	13.8	13.3
LIP-OPEN	14.0	15.3	14.6	13.3	13.5	13.1	13.0	13.9	12.8	**12.2**	13.1	12.4
TT-LOC	25.5	27.5	25.2	20.6	24.8	23.6	23.5	23.7	22.0	**19.1**	22.4	20.4
TT-OPEN	71.9	55.1	51.9	38.3	54.4	51.5	51.8	48.6	42.0	**36.8**	42.2	38.9
TB-LOC	55.7	47.4	45.2	31.9	44.0	41.5	41.4	40.2	34.2	**31.7**	35.1	33.1
TB-OPEN	53.2	50.1	47.8	32.7	44.8	43.1	43.0	44.2	34.7	**31.3**	35.6	32.3
VEL	16.8	17.0	17.0	11.6	14.8	14.3	14.3	15.3	12.7	**11.5**	13.0	11.8
GLOT	19.0	15.1	13.3	8.7	14.9	13.7	13.5	12.2	9.2	**8.4**	10.0	8.7
Phones	N/A	N/A	N/A	N/A	77.5	74.6	74.4	75.2	N/A	**60.1**	63.7	60.3

Figure 2 shows confusion matrices for all of the AFs, using the MLPP k-NN classifiers. The classes with the highest prior probabilities tend to dominate. While this is natural, it may be problematic and should be particularly so for large k. One possible way to mitigate this effect is to weight neighbors differently depending on their distances from the test vector, a possible improvement for future work.

5. References

[1] C. Browman and L. M. Goldstein, “Articulatory Phonology: An Overview,” *Phonetica*, vol. 49, pp. 155–180, 1992.

[2] L. Deng, G. Ramsay, and D. Sun, “Production Models as a Structural Basis for Automatic Speech Recognition,” *Speech Communication*, vol. 22, no. 2-3, pp. 93–111, 1997.

[3] K. Livescu et al., “Articulatory Feature-Based Methods for Acoustic and Audio-Visual Speech Recognition: Summary from the 2006 JHU Summer workshop,” *ICASSP*, 2007.

[4] K. Livescu, “Feature-Based Pronunciation Modeling for Automatic Speech Recognition,” Ph.D. Dissertation, Massachusetts Institute of Technology, 2005.

[5] C. Hu, X. Zhuang, and M. Hasegawa-Johnson, “FSM-Based Pronunciation Modeling Using Articulatory Phonological Code,” *Interspeech*, 2010.

[6] K. Kirchhoff, G. Fink, and G. Sagerer, “Conversational Speech Recognition Using Acoustic and Articulatory Input,” *ICASSP*, 2000.

[7] O. Çetin et al., “An Articulatory Feature-Based Tandem Approach and Factored Observation Modeling,” *ICASSP*, 2007.

[8] H. Hermansky, D. P. W. Ellis, and S. Sharma, “Tandem Connectionist Feature Extraction for Conventional HMM Systems,” *ICASSP*, 2000.

[9] T. Deselaers, G. Heigold, and H. Ney, “Speech Recognition with State-Based Nearest Neighbour Classifiers,” *Interspeech*, 2007.

[10] A. Andoni and P. Indyk, “Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions,” *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2006.

[11] K. Weinberger, “Large Margin Nearest Neighbors (LMNN) toolkit,” 2007. [Online]. Available: http://www.cse.wustl.edu/~kilian/code/files/LMNN.zip

[12] X. He and P. Niyogi, “Locality Preserving Projections,” *Advances in Neural Information Processing systems (NIPS)*, 2003.

[13] N. Morgan and H. A. Bourlard, “Neural networks for statistical recognition of continuous speech,” *Proceedings of the IEEE*, vol. 83, no. 5, pp. 742–772, 1995.

[14] S. Greenberg, J. Hollenback, and D. P. W. Ellis, “Insights into Spoken Language Gleaned from Phonetic Transcription of the Switchboard Corpus,” *ICSLP*, 1996.

[15] D. P. W. Ellis, “PLP, RASTA and MFCC in Matlab,” 2005. [Online]. Available: http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/

[16] L. van der Maaten, “Matlab Toolbox for Dimensionality Reduction (v0.7.2),” 2010. [Online]. Available: http://homepage.tudelft.nl/19j49/

[17] D. Johnson et al., “ICSI QuickNet software package,” http://www.icsi.berkeley.edu/Speech/qn.html, 2004.

[18] A. Subramanya and J. Bilmes, “The Semi-Supervised Switchboard Transcription Project,” *Interspeech*, 2009.