Measuring Unsupervised Acoustic Clustering through Phoneme Pair Merge-and-Split Tests

John Kominek, Alan W Black

Language Technologies Institute
Carnegie Mellon University
{jkominek,awb}@cs.cmu.edu

Abstract

Subphonetic discovery through segmental clustering is a central step in building a corpus-based synthesizer. To help decide what clustering algorithm to use we employed merge-and-split tests on English fricatives. Compared to reference of 2%, Gaussian EM achieved a misclassification rate of 6%, K-means 10%, while predictive CART trees performed poorly.

1. Introduction

Consider the problem of creating a speech synthesizer capable of multiple-dialects, e.g. a single voice able to speak all the major dialects of North America or of the British Isles. This ability does not yet exist. It requires a degree of control not currently available, and challenges the common practice of basing a system on conventional phoneme sets. To support multiple dialects in a single system, one must build on top of a more fine-grained set of elementary sounds.

The question arises: *which* elementary sounds to adopt as fundamental? Which accented phonemes, allophones, and subphonetic units are most distinguishing — hence useful? Since this question has no ready-made answer, the general solution is that some form of unsupervised learning is required to discover these elements, followed by testing in a functioning system. Listening tests are critical to reveal the effects of increasing unit granularity on coverage.

Two preliminary questions need to be addressed. Namely, *what* clustering algorithm to employ, and *how do you know* it is performing well? One solution is to construct new test sets from prior labeled data. Specifically, we merge many pairs of distinct fricatives – such a /s/ and /z/ – measuring how well various clustering algorithms perform at separation. Our evaluation suite comprises 28 merge-and-split tests.

This paper investigates 3 types of algorithms: standard K-means (using the open source program *pyCluster* [1]), CART tree clustering (using the open source program *wagon* [2]), and dual-Gaussian EM with full covariance matrices (re-implemented for this investigation). An upper bound on performance is provided by a set of eight reference models trained with knowledge of the actual phoneme identities. We find that Gaussian EM achieves an error rate of 6%, while the best K-means result is slightly over 10%. Furthermore, the consonant confusion matrices are generally compatible with human perceptual tests. These findings support and guide our agenda for extending acoustic discovery down a notch to the subphonetic level.

2. Related Work

The unsupervised discovery of elemental units of speech for corpus-bases TTS is an unexplored area. Precedents do exist for improving ASR performance using automatically derived sub-word units [3] along with a derived lexicon [4] (with mixed results). Multi-lingual speech recognition systems are often based on a superset of phonemes [5]. If one of the target languages suffers from a paucity of data, it is necessary to map models from a source language rich in resources. Doing this automatically requires a distance measure for cross-language phoneme comparison. In the work of [6] the phonemes of Afrikaans are mapped onto English, then cross-adapted.

Having a metric that can measure the distance between phone models is useful for dialect classification [7,8] and is potentially applicable to multi-lingual TTS (where currently phoneme mappings are assigned by hand [9]). One measure that has proved fruitful for establishing distances between phoneme classes is the Bhattacharyya distance [10,11].

For a TTS system, it is not enough to derive subphonetic units. One must also be able to predict the appropriate class from the input text. In Festival, this task is delegated to Classification and Regression Trees (CARTs) [2,12]. Since CARTs tackle the larger, joint problem of predictive acoustic clustering, an adequate solution here immediately solves the brunt of our problem. But, such a gift is not forthcoming.

In Section 3 we discuss the shortcomings of predictive clustering. Section 4 lays out a framework of possible approaches. Results and discussion follow in Section 5.

3. Shortcomings of Predictive Clustering

Nominally, Festival unit selection voices are phoneme-based. This is true in the sense that words in the pronunciation dictionary are described using symbols recognizable as IPA phonemes. Down in the details, though, units are selected from subphonetic clusters that are roughly equivalent to allophonic subtypes. These subphonetic clusters are learned from data and represented as CART trees. Why then not use the familiar Edinburgh speech tool *wagon*?

One shortcoming is that *wagon* tackles the problem of predictive clustering. In predictive clustering each unit is affiliated with a set of predictees — e.g. the surrounding phonetic context including syllable structure, phoneme features, Fo pattern, etc. These predictees are searched for dividing points (organized in a *yes-no* decision tree) that best explain acoustic differences between units. This makes the clustering dependent on the predictive value of the feature set. If a natural acoustic separation is not predictable from the available features then the clusters cannot be found.

A second shortcoming is that *wagon* is designed to generate balanced trees, which won't always match the data. For example in the single speaker database *rms_arctic* [13] there are 279 instances of /jh/ and 37 instances of /zh/. If these are merged and split into two clusters, the results are nearly balanced – as shown in the top half of Table 1 – leading to a large classification error of 127/266 = 47.7%. (The maximum error in a two-class problem is 50%.)

Unbalanced populations are always problematic, but high error rates persists even when the initial class sizes are nearly equal. The bottom half of Table 1 shows the case of /ch-sh/. Instead of clean separation there is large overlap between the clusters (/sh/ is split evenly), with a classification error of 253/543 = 46.6%. Such high error rates suggest trying non-predictive clustering algorithms.

phone	CL1	CL2	Total
JH	114	115	229
ZH	24	13	37
Total	138	128	268
CH	91	127	218
SH	163	162	325
Total	254	289	543

Table 1. Confusion matrix of two merge-split tests using wagon. Incorrect classification cells are highlighted.

4. Acoustic Distance Measures

Before running a classification algorithm several key options need defining. First is the tuple of features used to describe a frame of speech. This defines an embedding feature space which must have an assigned metric, allowing the distance between two points to be computed. The feature space may also undergo a data-dependent coordinate transformation.

Because a metric applies to a pair of points in the embedding space and speech is a trajectory (a sequence of vectors), the distance computation must be extended to deal with trajectories of unequal lengths. Comparing a pair of trajectories requires some form of time alignment.

4.1. Feature vectors and embedding spaces

We explored five feature vector variants, all related to the conventional specification: 12-dimensional fixed width mel frequency cepstral coefficients, derived from a 24 filterbank power spectrum with coverage up to 16 kHz, with 10 ms step size between frames, a Hamming window of 25 ms, and pre-emphasis of $1\text{-}0.97z^{-1}$. Voicing (in item 3) is a binary feature.

1. cepstral coefficients, 12-D.
2. cepstral coefficients + log power, 13-D.
3. cepstral coefficients + log power and voicing, 14-D.
4. cepstral coefficients + deltas, 24-D.
5. original log spectrum filterbank output, 24-D.

4.2. Coordinate space transformations

With speech data positioned in an embedding space, the simplest thing is to leave the vectors untouched. However, it is often advantageous to transform the points into a new coordinate system. Principal components analysis is a data-driven way to find a linear transformation that shifts, rotates, rescales, and orders the coordinate axes in way such that they are maximally aligned with the data's distribution [14,15]. If $\mathbf{X}$ is an dxn data matrix with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}=\mathbf{X}\mathbf{X}^t$, then the new representation is $\mathbf{Y} = \mathbf{P}(\mathbf{X}-\boldsymbol{\mu})$, with the rows of $\mathbf{P}$ being the principal components of $\mathbf{X}$. Also known as the Karhunen-Loéve transformation, the principal components are eigenvectors of the covariance matrix $\boldsymbol{\Sigma}$. Transforming $\mathbf{X}$ to $\mathbf{Y}$ diagonalizes the data's covariance, which can then be normalized so that the variance along each axis is unity.

The transformation $\mathbf{P}$ can be simplified to be a diagonal matrix that rescales the axes of $\mathbf{X}$ independently, without rotation. We use the term Zscale transformation to describe the multi-dimensional extension of the z-score. If the data component along one dimension x_i has mean μ_i and variance σ_i, then $y_i = (x_i - \mu_i) / \sigma_i$.

4.3. Distance metrics

When given random variable $\mathbf{X}$ the usual preferred choice for measuring the distance between two vectors is the Mahalanobis distance. This is equivalent to the Euclidean metric applied to the space $\mathbf{Y}$ after undergoing PCA.

Because of the extra flexibility provided by the software tool pyCluster, we ran K-means clustering under six different metrics, including Manhattan distance and four correlation-based measures. Given two vectors $\mathbf{x}$ and $\mathbf{y}$:

1. Euclidean distance. $d=[(x-y)^t(x-y)]^{1/2}$
2. Manhattan distance. $d=\sum|x_i-y_i|$
3. unentered correlation. $d_{uc}=\sum x_i y_i$
4. centered correlation. $d_{cc}=\sum(x_i-\mu_i)(y_i-\mu_i)$
5. absolute uncentered corr. $d=|d_{uc}|$
6. absolute correlation. $d=|d_{cc}|$

4.4. Vector extension

The K-means algorithm available in pyCluster is capable of handling fixed length time sequences of *scalar* quantities. Since speech frames are vectors, this is a problem. One way to overcome this limitation is to concatenate L frames to define a new single vector of length DxL, $V = (x_{11}, x_{12},... x_{1D}, x_{21} \ldots x_{LD})$. This vector extension procedure does not alter Manhattan distances, but does alter Euclidean distances.

4.5. Time Alignment

The formulas of section 4.3 require that units $\mathbf{x}$ and $\mathbf{y}$ are equal length. Vector sequences of unequal length need to be accommodated through time alignment. Two variants are dynamic time warping (dtw) and linear time warping (ltw). Linear warping maintains a proportional ratio between frames. If $\mathbf{x}$ has length m, and $\mathbf{y}$ length n, then two frames x_i and y_j map onto each other when j = min (n, round(i x n / m)).

The well known technique of dynamic time warping establishes a nonlinear mapping j=f(i) that minimizes the total path cost of matching $\mathbf{x}$ to $\mathbf{y}$. DTW abstracts away variations in time pattern. It accommodates, at little extra cost, sections of a phoneme that are sped up or drawn out. In contrast, ltw penalizes differences in the time curves, even when vectors follow identical paths. A TTS system should distinguish timing patterns that sound different, which is why Festival uses ltw when computing the distance matrix for clustering.

Figure 1 illustrates linear time warping between two phoneme units, projects onto the first two cepstral dimensions. Also shown is a rotated ellipse indicating a single Gaussian model for the data, and hence the PCA transform.

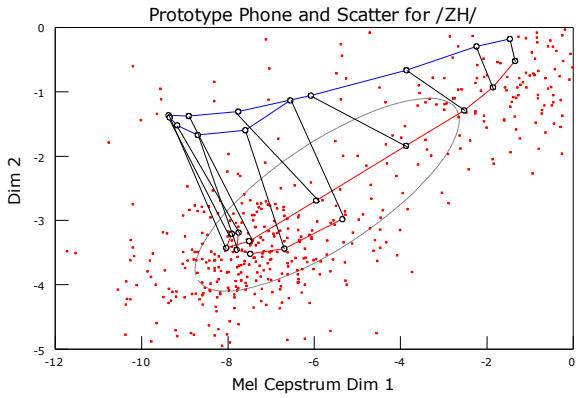

Figure 1*. Linear time warping between two unit trajectories. The ellipse indicates the PCA coordinate space transformation based on the phone class /zh/. Red dots are feature frames of /zh/.*

4.6. Cluster center possibilities

There are three possibilities for the cluster centroid: mean, median, and medoid. The medoid is the particular unit **x** whose total distance to all other points is minimum. This has a useful consequence. Because the medoid is determined by a distance cross-table, it is not necessary to know the actual data vectors. Computing mean and median centroids, by contrast, does require the original values. Like the medoid, the median will be a particular unit. The mean is a computed point, somewhere in the embedding space.

4.7. Comparison of model updating

The modeling framework just laid out is helpful for comparing various clustering algorithms. As evaluated here:

1. ***wagon***. Space is Zscale transformed according to the data of merged phone pairs, and then fixed. Uses a distance matrix only; cluster means are not explicitly updated. Is based on linear time warping.
2. ***K-means***. Space is PCA transformation of all speech data. Cluster means are updated with each iteration; distance metric is not. Is based on linear time warping.
3. ***Gaussian EM***. Space is PCA transformation of all speech data. Both cluster mean and distance metric are updated after each iteration (PCA). Time warping is not necessary.

4.8. Theoretical model separation

There are many measures of probabilistic model separation including K-L divergence and its symmetric counterpart, the "information radius." Another, Bhattacharyya distance [15] has been used extensively to obtain distances between single Gaussian phone models of different languages [10,11]. Figure 2 compares this estimate to empirical measurements.

$$D=\frac{1}{8}(\mu_2-\mu_1)^t[\frac{\Sigma_1+\Sigma_2}{2}]^{-1}(\mu_2-\mu_1)+\frac{1}{2}\ln\frac{|\frac{1}{2}(\Sigma_1+\Sigma_2)|}{\sqrt{|\Sigma_1||\Sigma_2|}} \quad (1)$$

The first term of eqn. 1 gives the model separability due to differences in means, relative to the average covariance, while the second term accounts for differences between the covariance matrices themselves.

5. Experimental Results

Instead of trying all possible phone pairs, we kept the number manageable by examining the natural grouping of alveolar-region fricatives and affricatives: /th, dh, s, z, sh, zh, ch, jh/. There are C(8,2) = 28 pairwise combinations. The speech database studied here is *rms_arctic*, available online [13].

5.1. Oracle reference vs Gaussian EM

For each test our reference point is a pair of Gaussian probability density models with full covariance matrices. These are "oracle" models because they are trained using the known class labels. The feature vectors are 12-D cepstra.

Though much simpler than a mixture-of-Gaussians HMM model, the overall pair-wise phoneme classification errors is small, only 2.6% and 6.0%. Table 2 shows the error percentages for 28 phone pair experiments. The oracle results are in the upper triangle and the learned results in the lower triangle. E.g. for /ch-sh/ the oracle models yield a total error of 10.7% and in the learned models this increases to 12.7%.

Table 2 *(below). Phone pair classification errors. Oracle models in upper triangle; learned models in lower triangle. Notice that the smallest phone class /zh/ is difficult to separate.*

	CH	DH	JH	S	SH	TH	Z	ZH
CH		0.7	11.9	0.6	10.7	0	0.4	1.57
DH	1.01		1.46	0.52	0	16.7	0.29	0
JH	21.7	1.69		0.54	5.05	0	0.4	4.89
S	0.6	0.56	.54		0.46	0.22	12.8	0.85
SH	12.7	0	5.23	0.52		0	0.45	10.2
TH	0	30.4	0	0.22	0		0.08	0
Z	0.49	.24	0.56	21.8	0.45	.08		0.66
ZH	20	34.6	38.7	5.99	11.9	0	3.79	

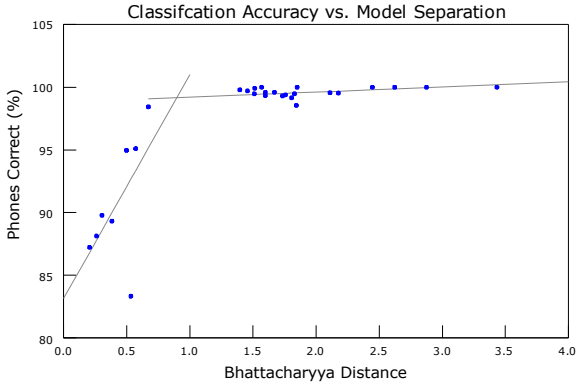

Figure 2. *Empirical vs theoretical model separation of oracle single Gaussian full covariance models.*

5.2. Comparison to perceptual experiments

In the classic consonants-in-noise confusability experiments of Miller and Nicely, they found that /th-dh/, /s-z/, and /sh-zh/ are maximally confusable pairs [16]. Since they did not test the affricatives, this is compatible with our results. The merge-split tests reveal that /th-dh/ and /s-z/ and /ch-jh/ are maximally confusable pairs, but that /sh-zh/ is not: /sh/ is most likely to be misclassified as /ch/; and /zh/ as /jh/.

Later experiments by Wang and Blinger included affricatives, with slightly differing results [17]. Among the eight phonemes considered here, they found that the post-alveolars coupled; i.e. /ch-sh/ and /jh-zh/ are confusable pairs.

Figure 3 offers insight by plotting the model means. Grey lines connect natural pairs that differ only in the voicing feature. Solid black lines point to the nearest model according to Bhattacharyya distance. The "distance between dots on paper" equals the Euclidean distance between model means projected onto the first two cepstral coefficients. Inspection of Figure 3 reveals, for example, how /sh/ might "consider" /ch/ to be its closest neighbor, instead of the voiced twin /zh/.

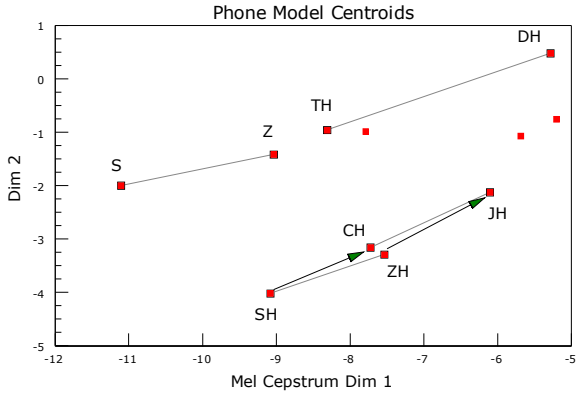

Figure 3*. Centroids of phonemes projected onto the first two cepstral dimensions. The gray lines connect unvoiced-voiced pairs. Arrows show other confusabilities.*

5.3. K-Means classification accuracy

Because K-Means clustering does not adaptively update the cluster's covariance matrix, it should not perform as well as 2-way Gaussian EM. This held true. We ran experiments under many different configurations and recorded a best error rate of 10.1%. Table 3 helps answer the question of which distance metric and centroid definition is optimal in the K-means algorithm configuration.

Distance Metric	Centroid Type		
	Mean	Median	Medoid
abs corr.	3752	4006	8677
abs uncntrd. corr.	3575	3734	8127
correlation	3672	3939	7734
uncentered corr.	3485	3635	6817
Euclidean	3389	3473	7878
Manhattan	3355	3559	8355
Total error count	21228	22346	47588

Table 3. K-Means separation using the same feature vectors as in Table 2. There are 33166 units in the database; the best result of 3355 wrong equals 10.1%. The embedding space is 12-D PCA melcep.

This shows that the best definition of a cluster centroid is arithmetic mean, closely followed by median, with medoid a distant third. This is just as well. Computation of the mean is O(n), while the median is O(n *log* n), and the medoid is $O(n^2)$.

The best choice of distance metric is less clear, but does favor Manhattan and Euclidean over correlation-based measures. Testing on datasets in which other feature vectors are used – for example the 24-D filterbank output – confirms this trend.

Distance Metric	Global Vector Space		
	Fbank 24	Zscale	PCA
abs correlation	10440	11148	4013
abs uncntrd. corr.	8938	8416	3768
correlation	10447	10649	3890
uncentered corr.	8935	8325	3675
Euclidean	7622	7965	3592
Manhattan	7312	7259	3682
Total error count	53694	53762	22620
	melcep 12	Zscale	PCA
Compare to	58272	26583	21228

Table 4. K-Means separation in the 24 dimensional filterbank vector space, as well as under Zscale and PCA transformations. The best error rate is 10.8%. The bottom row compares to 12-D cepstral vectors.

In Table 4 we see something interesting that is not easily explainable. Normalizing the data variance along each axis of the original vector space (applying a Zscale transformation) helps in the cepstral domain, but does not help at all in the filterbank domain. Comparison to an analysis that uses the Fisher linear discriminant may prove clarifying.

6. Conclusion

Our phoneme pair merge-and-split experiments reveal that Gaussian EM (one per cluster) performs best. K-Means is adequate when operating in PCA space, using mean centroids and a Euclidean or Manhattan metric. CART trees didn't not do well on blind separation. This deserves more investigation.

Our next step is to repeat this procedure for vowels and other consonant groups. Following that, examine known phonological rules of English such as described in [18]. This will take us closer towards reliable subphonetic discovery.

7. Acknowledgments

This work is in part supported by the US National Science Foundation under grant number 0415021 “SPICE: Speech Processing — Interactive Creation and Evaluation Toolkit for new Languages.” Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

8. References

[1] de Hoon, M, Imoto, *et al*. Open Source Clustering Software, *Bioinformatics*, vol. 20(9): pp. 1453-54, 2004.

[2] Edinburgh Speech Tools Library. www.cstr.edu.ac.uk/projects/speech_tools.

[3] Bacchiani, M. *Speech Recognition System Design Based on Automatically Derived Units*, Ph.D. dissertation, Technical University of Eindhoven, 1994.

[4] Singh, R., Raj, B., Stern, R. M. “Automatic Generation of Phone Sets and Lexical Transcriptions,” *IEEE Trans. on Speech and Audio Processing*, vol. 10(2), pp. 89-99, Feb. 2002.

[5] Schultz, T., Waibel, A. “Fast Bootstrapping of LVCSR Systems with Multilingual Phoneme Sets,” *Proceedings of the EuroSpeech*, Rhodes, Greece, pp. 371-74, 1997.

[6] Sooful, J. *Automated Phoneme Mapping for Cross-Language Speech Recognition*, M.Eng Thesis, University of Pretoria, 2004.

[7] Miller, D., Trischitta, J. “Statistical Dialect Classification Based on mean Phonetic Features,” *Proceedings of ICSLP-1996*, Philadelphia, USA, pp. 2025-27.

[8] Yamamoto, K., Nakaguwa, S. “Relationship Among Speaking Style, Inter-Phoneme's Distance and Speech Recognition Performance,” *Proceedings of ICSLP-2000*, Beijing, China.

[9] Badino, L., Barolo, C., Quazza, S. “Language Independent phoneme Mapping for Foreign TTS,” *5th ISCA Speech Synthesis Workshop*, Pittsburgh, USA, 2004, pp. 217-18.

[10] Mak, B., Barnard, E. “Phone Clustering Using the Bhattacharyya Distance,” *Proceedings of ICSLP-1996*, Philadelphia, USA, pp. 2005-8.

[11] Sooful, J., Botha, E. “Comparison of Acoustic Distance measure for Automatic Cross-Language Phoneme Mapping,” *Proceedings of ICSLP-2002*, Denver, USA, pp. 521-24.

[12] Black, A., Taylor, P. “Automatically Clustering Similar Units for Unit Selection in Speech Synthesis,” *Proceedings of the EuroSpeech*, Rhodes, Greece, pp. 601-4, 1997.

[13] Kominek, J. and Black, A. “The CMU ARCTIC Speech Databases,” *5th ISCA Speech Synthesis Workshop*, Pittsburgh, USA, 2004, pp. 223-24.

[14] Schlens, J. *A Tutorial on Principal Component Analysis,* www.snl.salk.edu/~chlens/notes.html.

[15] Duda, R., Hart, P., Stork, D. *Pattern Classification,* John Wiley & Sons, 2001.

[16] Miller, G., Nicely, P. “An Analysis of Perceptual Confusions among some English Consonants,” J*ournal of Acoustical Society of America*, vol 27(2), 1955.

[17] Wang, M., Bilger, R. “Consonant Confusions in Noise,” Journal *of Acoustical Society of America*, vol 54(5), 1973.

[18] Ladefoged, P. *A Course in Phonetics*, Harcourt, Brace, Jovanovich, 2001.