Kobe University Repository : Kernel

PDF issue: 2026-11-06

Sibilant Representation Using MFCCs and GMMs

Pinter, Gabor

(Citation)
神戸大学国際コミュニケーションセンター論集,11:72-84

(Issue Date)
2014

(Resource Type)
departmental bulletin paper

(Version)
Version of Record

(JaLCDOI)
https://doi.org/10.24546/81008805

(URL)
https://hdl.handle.net/20.500.14094/81008805

KOBE UNIVERSITY

Sibilant Representation Using MFCCs and GMMs

Gábor Pintér[1]

1. Overview

The task of representing speech sounds entails a non-trivial, interdisciplinary challenge, with no single obvious, or ultimately correct solution. Formulating speech representations can be compared to the construction of physical research models. For example, the aerodynamic characteristics of an airplane can be modelled using a solid wooden model mounted in a wind tunnel. The same model, however, would perform poorly in a real flight test because it lacks many critical features required for flying. For this latter purpose a powered, dynamically scaled free-flight model would suit better. Alternatively, computational models or simulators can also be used to simulate the behavior of the plane under extreme situations, such as flight in the middle of a tornado, or when one of the wings is missing. All these models are abstractions in the sense that they ignore certain details of the original plane in order to investigate some other characteristics. The nature of simplification, and the characteristics highlighted by the resulting model are largely determined by the purpose of the inquiry.

Table 1 Approaches to segment representation

Domain	*General goal*	*Examples*
Phonology	to describe/explain competence	· binary features (Chomsky & Halle, 1968) · autosegments (Goldsmith, 1979)
Psycholinguistics	to describe/explain human performance	· beads on a string phonemes (e.g., TRACE: McClelland & Elman, 1986)
Automatic Speech Recognition (ASR)	to improve machine performance	· Gaussian Mixture Models (numerical coefficients) for sub-phonemic units

Representing speech sounds is similar to modelling airplanes in several respects. Different domains require different representations. The fields of phonology, psycholinguistics and speech recognition all rely on abstract representation of speech sounds, but they operate in rather divergent paradigms. Phonology defines mental representations based on *theoretical assumptions* (e.g., economy of representation) aiming to cover among others typological variations, diachronic changes, synchronic phenomena. Psycholinguistics is less interested in abstract principles: it aims to develop models that explain certain aspects of *human performance* and tests them by experiments, for example, how carefully synthesized audio stimuli are perceived, or how fast our reactions are in various perceptual tasks. In contrast, the field of Automatic Speech Recognition (ASR) focuses on *machine performance*. While ASR systems borrow some ideas from phonology (e.g., phonemes) and psycholinguistics (e.g., neural networks), they mainly strive to develop fast and accurate speech-to-text algorithms. Refer to Table 1 for a brief comparison of these approaches.

[1] School of Languages and Communication, Kobe University, g-pinter@port.kobe-u.ac.jp
This work was supported by JSPS KAKENHI Grant Number 26284059, 26770141.

The differences in terms of goals and foci have led to rather different, often incompatible ideas concerning the abstract representation of speech sounds. The goal of this paper is to investigate how Gaussian Mixture Models (GMMs) with Mel-Frequency Cepstral Coefficients (MFCCs) can represent sibilant sounds. While GMMs and MFCCs are used extensively in digital signal processing and ASR settings, they are less common in circles of phonology or phonetics. They have some interesting properties that could help filling the gap between strictly theoretical approaches and performance-oriented statistical models. Besides explaining the potential of a GMM-based approach to phoneme representation, this paper also aims to contribute to a wider understanding of these heavily technical tools among audiences with less technical background in order to make them more accessible for those who work in the fields of phonology and psycholinguistics.

2. Calculating Mel-Frequency Cepstral Coefficients

The following description of mel-frequency cepstral coefficients (MFCC) is from Wikipedia:

> "In sound processing, the mel-frequency cepstrum (MFC) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a non-linear mel scale of frequency. Mel-frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an MFC."
>
> *Wikipedia, retrieved on 28/02/2015*

Probably this description does not provide much help to clarify the concept for those who are untrained in digital signal processing. It may look formidable, but the actual procedure of calculating MFCCs is not that complicated. The rest of this section is a step-by-step description of calculating MFCC features.

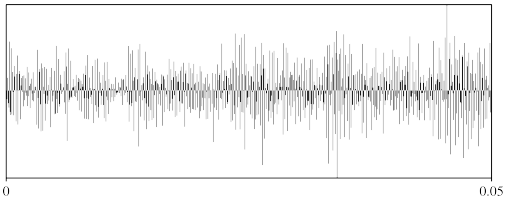


Fig. 1 Sound wave from sibilant [s]

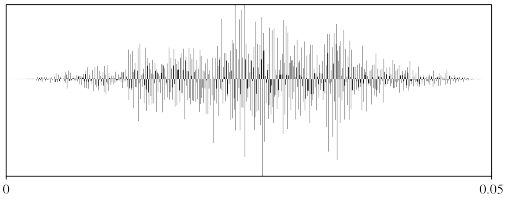


Fig. 2 Hanning windowed sound wave

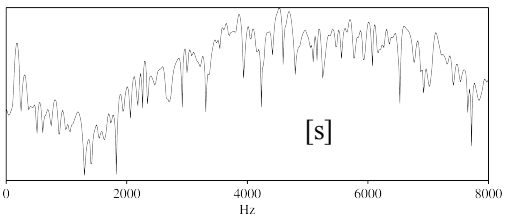


Fig. 3 Power spectrum calculated using FFT

Most speech analyses starts by slicing up the speech into 5 to 50ms long stretches. Fig. 1 shows a 50ms long stretch taken from the middle part of a Japanese sibilant [s]. In order to see the frequency components of this time-varying signal we need to transform the data from the time domain to the frequency domain. To put it very simply, if the horizontal axis is time, the plot represents the *time domain*, if the horizontal axis is frequency, we are looking at the *frequency domain*. The transformation that is commonly used to convert signals from the time domain to the frequency domain is called the Fast Fourier Transformation (FFT). For technical reasons FFTs are usually applied to a windowed signals. Note in Fig. 2 how the edges are converging to zero—after applying a Hanning window. The plot obtained using FFT is called the power spectrum, or spectral envelop. As displayed in Fig. 3, the spectral envelop of [s] shows greater intensities at higher frequencies, approximately over 4000Hz.

While power spectrums are greatly useful in speech analysis, they also have some shortcomings. First, spectral envelopes do not reflect human perception faithfully. On one hand, human perception is not linear on the

frequency domain, but logarithmic. For instance, a change of 100Hz is perceived as octave if changing from 100Hz to 200Hz, but perceived as a semitone from 1700Hz to 1800Hz. On the other hand, human perception is not as sensitive to spectral details as some hairy spectral envelops suggest. This leads to the second point. The power spectrum offers too much details, which can be problematic from a computational point of view as computational resources are often limited. Both the perceptual and the information technological problem can be addressed by *Mel-frequency cepstral coefficients* (MFCCs).

MFCCs rely heavily on the *mel scale*, which approximates human perceptual performance more closely than the physical frequency scale. Calculating MFCCs can be imagined as applying a series of filters on the power spectrum. The filters zero out most frequency values except for within a small triangle. Values within these triangles are added up. Fig. 4 displays a *filter bank*: a series of mel-filters. Although the widths of filtering triangles seem getting wider at higher frequencies, on the mel scale they are spaced evenly. It is typical to filter the power spectrum with 24 mel channels, resulting in 24 mel coefficients.

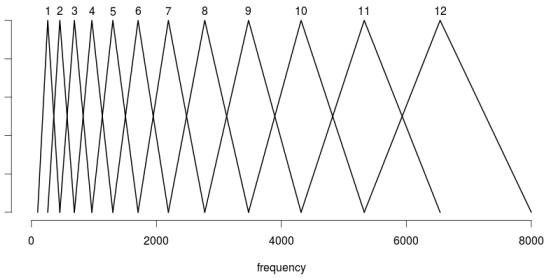


Fig.4 Mel-filter bank with 12 filter channels

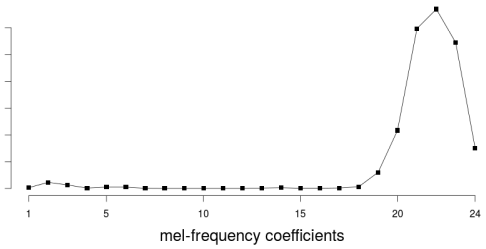


Fig.5 Mel frequency coefficients

The mel-filtering is followed by taking the log and the discrete cosine transform (DCT) of the values. The DCT is applied in order to separate the source and the filter characteristics of speech. It is common to decompose speech into its source and filtering components. The source component associated with pitch, loudness and phonation types such as whispery voice. The filtering component is related to the articulatory information in the speech signal. The separation of the two can be practical in speech recognition, as ideally pitch and phonation types should not influence the verbal content of our utterances, hence it should have no effect on recognition performance.[2] The DCT converts the signal from the frequency domain to the quefrency domain. Following the frequeny-quefrency analogy, the spectrum on the quefrency domain is called the cepstrum. The source component, which is mostly irrelevant, can be easily filtered out from the DCT by ignoring the upper half of the

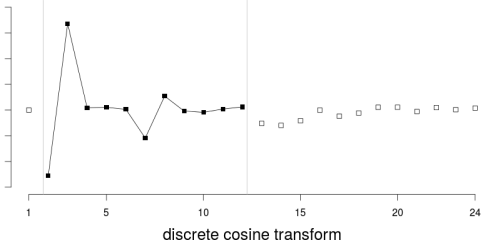


Fig.6 DCT output: empty boxes represents coefficients that are usually left out

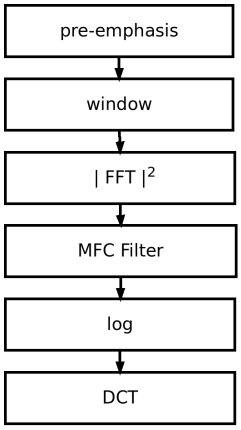


Fig.7 MFCC pipeline

[2] Unfortunately source characteristics can greatly influence speech recognition accuracy. For example, ASR systems show deteriorated performance with female or child speakers due to higher pitch. Also, specially trained systems are needed for whispery voice.

resulting numeric array. Also, the first component of the DCT is often left out—and replaced by the power calculated over the spectral envelop. The steps of the feature extraction pipeline is summarized in Fig. 7. The first step, pre-emphasis, was not discussed above. It is applied in order to model the radiation characteristics of the lips. An in-depth explanations of the steps involved calculating MFCCs and the rationale behind them can be found among others in Holmes & Holmes (2001:160–164).

3. Gaussian Mixture Models

Calculating MFCCs over speech segments in itself is neither a difficult nor a challenging task in itself. The real question is how these coefficients could serve as meaningful representation of speech. As discussed in the introduction, different task domains require different representations. MFCC representations were developed and typically used in statistical, computational models of speech recognition. They are often used as building blocks of with Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs). This section briefly explains how MFCCs are used in GMMs.

3.1 Multivariate Probability Distributions

Speech recognition can be conceptualized as a categorization task: the incoming signal has to be categorized into one of the available phoneme, tri-phone, syllable or word categories. The categorization is based on previous observations. The observations are stored in the form of numerical data. As an example, we can imagine the task of gender categorization (i.e. man versus woman) based on observation about body height.[3] Numerical data of body height has a symmetrical bell-shaped distribution called the *normal* or *Gaussian* distribution. Fig. 8 displays a hypothetical distribution of body height. Based on previous observation we can assign probability values to observations. In Fig. 8 a body height of 167cm has the probability value of 0.312. This means that selecting a sample with 167cm from this given population has circa 31.2% of chance. A very simple

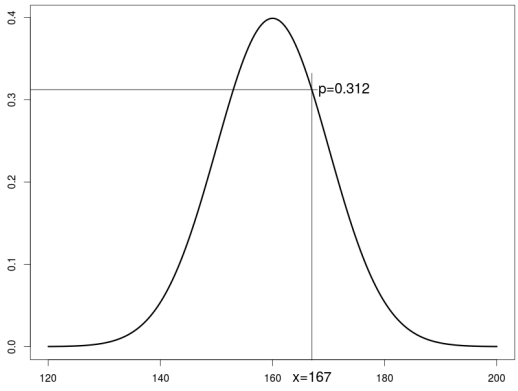


Fig. 8 Gaussian distribution

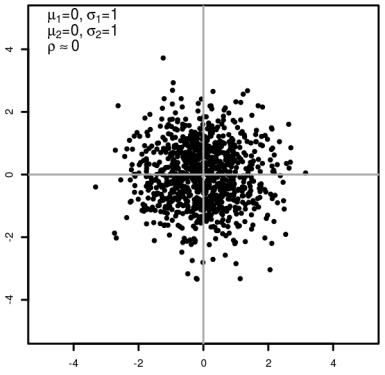


Fig. 9 Two random variables with no significant correlation

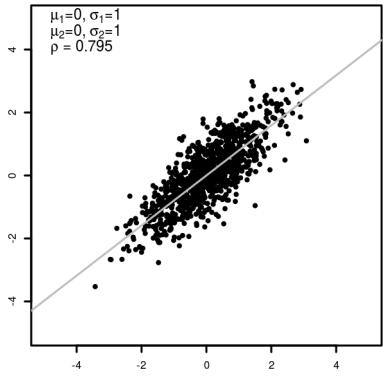


Fig. 10 Two random variables with strong correlation

[3] A very similar example demonstrating probability distribution can be found in Araki (2007:82).

categorizer would just compare these probabilities to decide if a body height value belongs to the male or the female population. In order to calculate the probabilities we just need to know the mean and the standard variation (SD) of height values for men and women.[4]

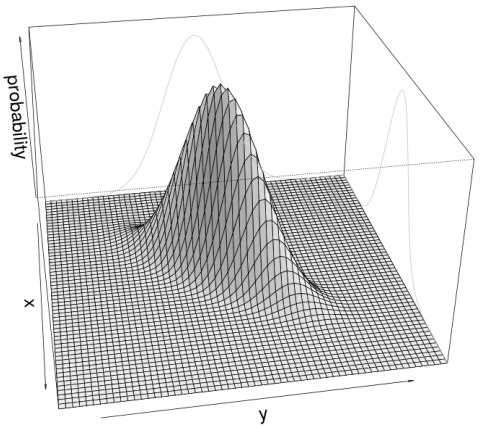


Fig. 11 Probability density function for two correlating variables

In most cases observations are not single valued. For example, our gender classifier could use other measurements, such as body weight or BMI index. These measurements in technical terms are called *random variables* or *independent variables*. The use of multiple random variables introduces a further component to the problem of calculating probabilities. Probabilities from a single Gaussian distribution can be calculated using the mean and the SD. However the probability distribution of two or more variables cannot be simply characterized by means and SDs. Means and SDs can only describe symmetrical distributions, such as in Fig. 9. This, however, is rarely the case. Returning to our gender classifier, most certainly we find correlation between body height and body weight. As a rough generalization it can be said that taller people tend to be heavier. This kind of correlations can be identified easily if represented visually. For example, in Fig. 10 the correlation between the two variables is apparent. Identifying values of correlation is important, because they are needed to calculate probabilities. Without the correlation term, only orthogonal, non-correlating variables, and symmetrical distributions can be described.

If the variables making up our observations are clustering around some mean values, their distribution is called *multivariate Gaussian distribution* or *multivariate normal distribution*. The probability for observations in multivariate cases are calculated using the means, the SDs for each variable, together with their pairwise correlation. Fig. 11 depicts a hypothetical bivariate Gaussian distribution in a fancy 3D perspective style. Probability densities for the individual variables are plotted against the "walls" at the back and on the right side. The individual variables also have Gaussian distributions.

Note that due to practical, and demonstrative considerations the multivariate examples and figures here are restricted to two dimensions, but in real-life situations classifier can work with dimension up to several thousands. The logic of classification, however, is the same as for the univariate case. Observation probabilities are calculated for each of the categories, and the observation is assigned to the category for which it has the highest probability. It is common to refer to the parameters (i.e. means, SDs, correlation matrices) that describe a category by the term *model*. So, for example, we can talk about statistical models of males and females. Likewise, Gaussian Mixture Models are statistical models that use Gaussian mixtures. The concept of mixtures is explained in the next subsection.

[4] This is a rather naïve approach. A more sophisticated one would consider the man-woman ratio as well.

3.2 Multivariate Multimodal Probability Distributions

Another way to extend the Gaussian model—besides increasing the dimensionality of observations—is to describe the data of random variable with multiple Gaussians distributions. Multiple components are needed if the distribution of the data has multiple *modes*, or so to say peaks. For example, the distribution displayed in Fig. 12 can be only poorly approximated with a single Gaussian distribution. The components of the composite distribution are called mixtures, the resulting model is a mixture model. Hence, the term Gaussian Mixture Model reflects the concept of representing random variables using mixtures of Gaussian distributions. The parameters of the mixture model consist of the means, the SDs and the weights of the individual mixtures. The probability value for an observations corresponds to the weighted sum of the probabilities calculated for the observation in each mixture component.

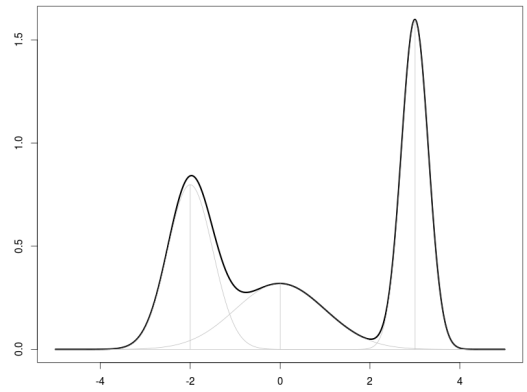


Fig. 12 A multimodal distribution with three Gaussian mixture components

With real life data, it is common that observations are described by more than one continuous variable (i.e. multivariate), and the variables have multimodal distributions. Fig. 13 demonstrates a hypothetical bimodal, bivariate distribution. Both variables have 2 modes, as apparent from the marginal distributions on the walls of the plot.

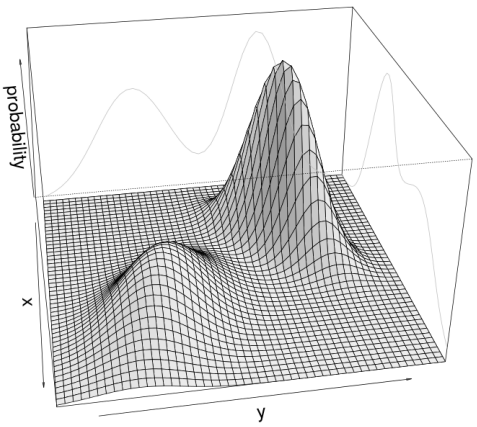


Fig. 13 A Gaussian mixture model with bimodal bivariate distribution

Although the mathematics gets complicated as the number of variables and the number of modes within the variables increases, the basic concepts are the same as in the simplest univariate, unimodal case. Categories are represented by models, or model parameters. The classification involves calculating probabilities for observations in each model. The observation is assigned to the model in which it has the highest probability score.

3.3 MFCC meets GMM

By now it is probably clear how MFCCs can fit into mixture models. MFC coefficients are treated as observations. Each channel in the MFCC correspond to a random variable for the mixture model. In case of speech data, it is common to use MFCC values not only for a given stretch of speech (*frame*), but calculate the difference between current and previous feature vectors (*delta*), or calculate the difference between the current and the previous differences (*delta-delta*). Calculating with 13 basic mel-features, and appending the deltas and delta-deltas results in a 39 value-long feature vector.

There is a very important aspect of GMMs that has not been discussed yet. So far, the parameters of the models were considered to be given. In reality, however, these parameters have to be estimated. Parameter estimation is a complex, time and computation consuming process. It is called the *model training*. The selection of the training algorithms, the preparation of the input data, feature selection are all complex issues exceeding the limits of this article. For more technical details the reader is advised to consult Holmes & Holmes (2001) or Jelinek (1997).

4. GMM classifier for Japanese sibilants

In order to demonstrate how GMMs can classify phonemes, a rather simple, [s] and [ʃ] discrimination task was chosen. Since GMMs cannot model temporal characteristics—such as formant transitions—well, a pair of segments were chosen that can be discriminated by non-dynamic acoustic features. Although formant transitions can play an important role in the discrimination of sibilants, Japanese listeners are reportedly capable of telling [s] and [ʃ] apart based solely on the noise components of the sibilants (Takeyasu, 2009). Since characteristics of the friction noise in sibilants—and generally in fricatives—are reflected in spectral differences, MFCC-based GMMs are expected to perform well in this task.

4.1 Stimuli

The speech data was constructed by sampling a bigger speech database holding data for 65 undergraduate Japanese students. As a part of an assessment of English language proficiency and speech fluency, students were asked to read out words on the screen, and give a short talk about a given topic. The test was carried out in a computer classroom, using headphones and mouthpiece microphones. Fig. 14 demonstrates that the speech data contains various environmental noises, clipping, and background speech.

For the training set, 23 English words—as displayed in Table 2— were chosen starting with either [s] or [ʃ]. Each word was spoken by 5 different speakers, the training set involved utterances from every speaker.

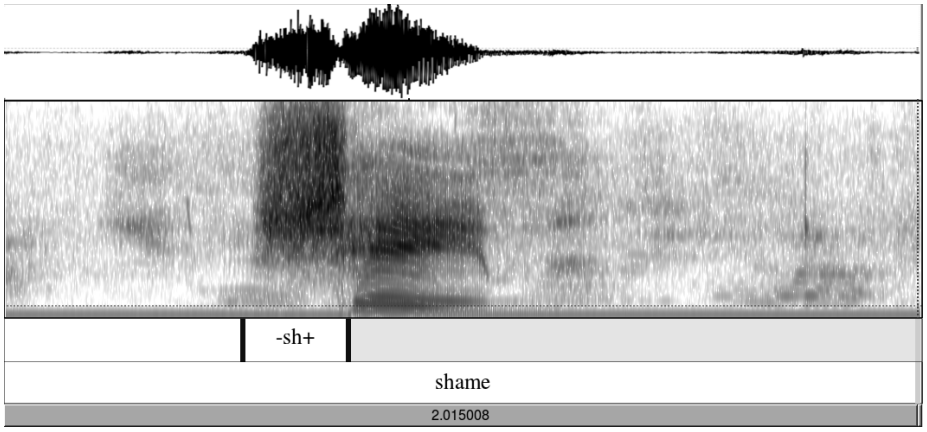


Fig. 14 Aligned target word sample—Praat window

Table 2 Target words

sad	*shine*
salt	*ship*
sand	*shirt*
save	*shock*
saw	*shoe*
seat	*shoot*
sent	*shop*
shack	*short*
shame	*sip*
shark	*sit*
shave	*soon*
sheet	

Segment labels were created only for the sibilant segments. In case of mispronunciation, the actual pronunciation was transcribed. For example, if '*sip*' was pronounced as [ʃip] the label '`-sh+`' was used instead of '`-s+`'.

4.2 Training

The training set consisted of 49 [s], and 66 [ʃ] sounds. For each sibilant the middle 50ms part was extracted and 24 MFC coefficients were calculated using a Python script. The GMM models for [s] and [ʃ] were trained over these 24 MFC coefficients ignoring delta and delta-delta values. For the training and the evaluation the `sklearn.mixture.GMM` Python module was used from the opensource machine learning library named *scikit-learn*. The plots in Fig. 15 describe distributions of the 12 coefficients. It can be seen that mel channels with higher index contribute less to the [s]-[ʃ] distinction. The distributions are getting similar with higher dimensions. This is one reason why higher dimensions in MFCC vectors are often left out in resource-critical ASR systems.

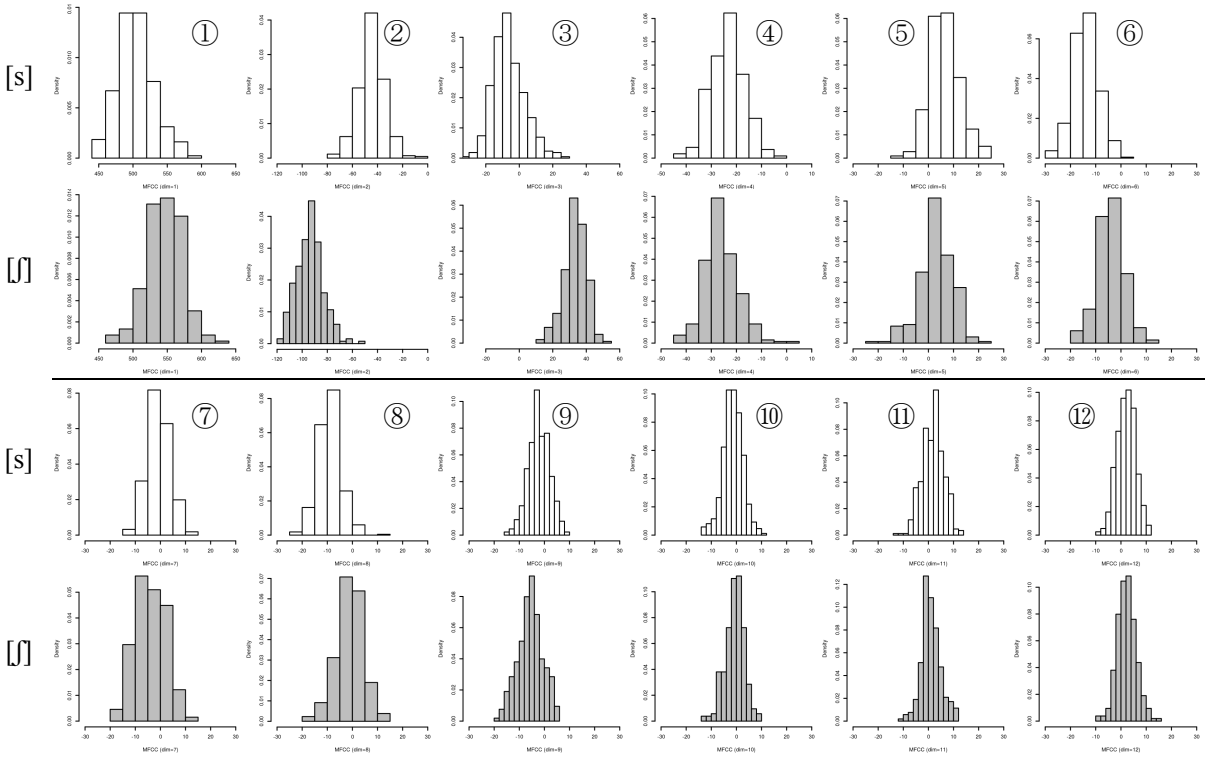


Fig. 15 Distribution plots for the first 12 MFCCs. Plots for [s] (white bars) and [ʃ] (filled bars) are paired vertically. The range of horizontal axes in the plots are fixed across pairs.

A series of mixture models were trained by permuting the following two parameters. First, the number of Gaussian mixtures were manipulated. Although Fig. 14 suggests that a single Gaussian can more-or-less correctly describe the distributions of training data coefficients,[5] GMMs were trained with 1, 2, 3, 4, 5, 10, and 20 mixtures. Second, *spherical*, *diagonal* and *full* covariance matrices were created and combined with the different number of mixtures. In sum, 21 GMMs were trained for each sibilant.

4.3 Test results and discussion

For testing purposes 20-20 words with [s] and [ʃ] segments were extracted from the speech corpus. There was no overlap between the items of the training and the test set. The test set was labelled and processed similarly to the training set. Probability values for labels [s] and [ʃ] were calculated by scoring the

[5] This is an overstatement. Model accuracy cannot be estimated reliably without knowing the co-variation between the variables.

feature vectors of the test data using the trained GMMs. The label for the GMM with the higher probability score was assigned to each test data point. Accuracy of the classification was calculated by dividing the sum of correct predictions by the size of the test set. The accuracy values for test and training sets are summarized in Figures 16-18.

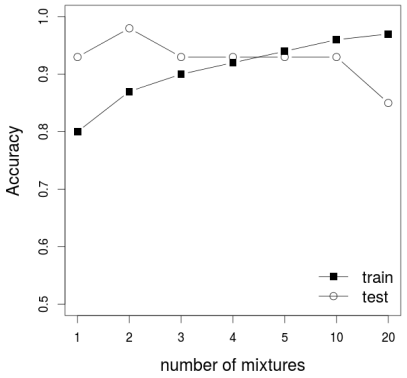


Fig. 16 Accuracy with spherical covariance

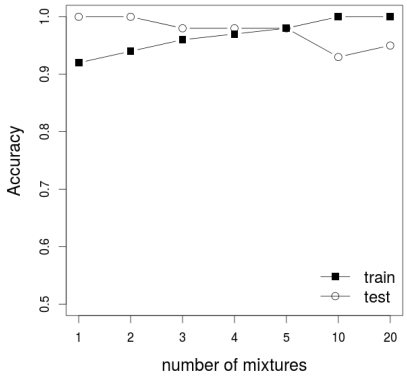


Fig. 17 Accuracy with diagonal covariance matrix

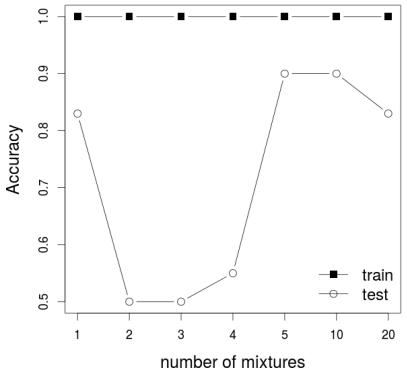


Fig. 18 Accuracy with full covariance matrix

It is clear from Figures 16-18 that increasing in the number of Gaussian mixtures increases accuracy in the training set. This tendency is not explicit for full covariance models as they reach 100% accuracy with a single mixture. In case of test sets, however, full covariance models do not benefit from an increase in the number of mixtures. Accuracy drops to chance level with 2 mixtures. The reason for this is that with more mixtures GMMs do not learn the phoneme category, but memorize the training data with all the minute details. This results in better performance with the training set, but leads to loss of generalization capacity, which is needed to classify unseen data. This explains why a simple spherical covariance model with 2 mixtures does not perform particularly well on the training set, but reaches accuracy close to 100% in the test set. Also, the corresponding full covariance model with 2 mixtures discriminates sibilants perfectly in the training set, while it has the performance of a random guesser in the test set. Considering the combined performance of training and test sets, it seems that GMMs with 5 mixtures and diagonal covariance provide the best performance for the sibilant discrimination task. It is needless to say that this small-scale experiment is just a demonstration, it does not allow for far-reaching generalizations.

5. Theoretical Considerations

After briefly describing MFCCs and GMMs, and demonstrating how sibilants can be represented and classified with their help, an important question remains. What relevance do these ASR oriented techniques have for phonology and psycholinguistics? Do they have the potential as a representational tool to bring phonology, psychology, and automatic speech recognition closer together? Although a detailed discussion is beyond the scope of this article, the answer seems to be positive.

GMMs provide an interesting, hybrid approach to phonological representation as they combine symbolic and non-symbolic features. One of the greatest dichotomies between models of representation is the symbolic versus non-symbolic divide (Clark, 1989). Most phonological theories use symbolic representation. Syllables, phonemes and features, whether they are single- or multi-valued, are inherently

symbolic. This means that in phonological descriptions and operations their reference to linguistic entities is explicit. For non-symbolic representations this link is not apparent. Non-symbolic representation can be illustrated through the process how someone recognizes her own grandmother. The symbolic approach would assume a dedicated neuron which is responsible for recognizing grandma: the grandma neuron. In reality, there is no such thing. The percept of a person comes from the detection of various features, from activations of thousands of neurons. This type of representation is non-symbolic, because it is present in the form of activation levels distributed all across the system. For the same reason GMMs are also non-symbolic. In the sibilant example, there are no clear acoustic or perceptual correlations to individual MFC coefficients. The statistical model in its whole, with all the probability values together, provide a representation for a segment.

The dual nature of GMMs comes from the fact, that while stochastic models are inherently non-symbolic, the linguistic entities they model are symbolic in nature. Of course, putting a label on a statistical model creates only a very superficial link between symbolic and non-symbolic systems. In order to make this link more meaningful it has to be demonstrated that GMMs have relevance both at symbolic and non-symbolic levels. As for the symbolic part, this requirement is already satisfied to a certain extent as GMMs usually model phonemes, which are symbols of phonology. Of course, treating phonemes as single, monolithic entities is a somewhat outdated phonological view. Actually, there are attempts to implement various types of features and inventories inspired by phonological theories,[6] but the real challenge with these approaches is to retain speech recognition accuracy while using phonological sub-systems.

The non-symbolic association of GMMs could be strengthened by interfacing more closely to psycholinguistics: GMMs could be improved in directions that help model various aspects of human performance. Being a tool with the purpose of recognizing and classifying speech is a feature that inherently brings a human trait into the system. Human-like speech recognition is, however, more like the holy grail of the field, rather than a feature for references. Presumably, convergence between human and machine performances at some less ambitious levels, such as phoneme confusion (cf. Peláez-Moreno et al. 2010) or perceptual similarity, could be the key to a global breakthrough in ASR performance. Returning to the sibilant example above, it could be a sensible extension of the research to compare how GMMs and humans perceive a synthesized S-SH continuum (Pinter 2007). The final goal would be to modify features of the sibilant GMMs in a way that the probabilities calculated for competing sibilant GMMs could be linked to the sigmoid discriminant function produced by human listeners. This may sound like a simple task, but it raises several questions. First, features for GMMs have to be reconsidered because they could greatly influence model performance. There are several acoustic features, such as center of gravity, and second formant onset (Jassem, 1965; Fujisaki & Kunisaki, 1978; Mann & Repp, 1980; Laver, 1994:206–263; Hirai et al., 2005) that can possibly mimic human perception better than MFCCs. Second, there is no general consensus about how probability values from GMMs could define a sigmoid discriminant function. Third, it is not clear if [s] and [ʃ] should be modelled as two monolithic unit, or

[6] An uncomprehensive list of phonologically motivated ASR systems would involve Waltrous & Shastri (1987) using binary phonological features; Deng (1997)'s experiment with autosegmental representation; Ahern (1999)'s implementation of unary features in the spirit of Government Phonology; Olaso & Torres (2010)'s approach to model articulatory features.

rather—following phonological approaches—describe the post-alveolar [ʃ] as the sibilant [s] with an additional post-alveolar (or palatal) feature. Accordingly, there is plenty of room for improvement at both the phonemic/symbolic and the phonetic/non-symbolic levels. Finding GMM representations that satisfy both symbolic and non-symbolic requirements forms the topic of future research.

In sum, it can be claimed that GMMs are promising representational tools at the hand of speech scientists, having the potential to combine symbolic and non-symbolic aspects of speech. They can be the cornerstones for those research projects that aim to bring together psycholinguistic, phonological and computational models of speech perception and speech recognition.

References

Ahern, S. (1999). *A government phonology approach to automatic speech recognition*. Unpublished master's thesis, The University of Edinburgh.

Araki, M. (2007). *Furii sofutode tsukuru onsei ninshiki shisutemu: patān ninshiki, kikaigakushū no shoho kara taiwa shisutemu made*. Tokyo: Morikita Press.

Chomsky, N., & Halle, M. (1968). *The sound pattern of English*. New York: Harper & Row.

Clark, A. (1989). *Microcognition: philosophy, cognitive science, and parallel distributed processing*. Cambridge, Mass.: MIT Press.

Deng, L. (1997). Autosegmental representation of phonological units of speech and its phonetic interface. *Speech Communication, 23*(3), 211–222.

Fujisaki, H., & Kunisaki, O. (1978). Analysis, recognition and perception of voiceless fricative consonants in Japanese. *IEEE ASSP, 26*(1), 21–27.

Goldsmith, J. A. (1979). *Autosegmental phonology.* New York: Garland.

Hirai, S., Yasu, K., Arai, T., & Iitaka, K. (2005). Acoustic cues in fricative perception for Japanese native speakers. *IEICE technical report*, *104*(696), 25–30.

Jassem, Wiktor (1965). The formants of fricative consonants. *Language and Speech*, *8*, 1–16.

Laver, J. (1994). *Principles in phonetics*. Cambridge: Cambridge University Press.

Mann, V., & Repp, B. (1980). Influence of vocalic context on perception of the [s]-[ʃ] distinction. *Perception and Psychophysics*, *28*(3), 213–228.

McClelland, J. L., & Elman, J. L. (1986). The TRACE model of speech recognition. *Cognitive Psychology, 18*, 1–86.

Olaso, J. M., & Torres, M. I. (2010). Integrating Phonological Knowledge in ASR Systems for Spanish Language. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications Lecture Notes in Computer Science, 6419*, 136–143.

Peláez-Moreno, C., García-Moral, A. I., & Valverde-Albacete, F. J. (2010). Analyzing phonetic confusions using formal concept analysis. *The Journal of the Acoustic Society of America, 128*, 1377–1390.

Pinter, G. (2007). The interaction of contextual vowels and formant transitions in Japanese fricative perception. *315th Regular Meeting of the Phonetic Society of Japan*, Tokyo University.

Takeyasu, H. (2009). Masatsu no sokuon ni okeru shūhasū tokusei no eikyō. *Phonological Studies*, *12*, 31–38.

Waltrous, R. L., & Shastri, L. (1987). Learning phonetic features using connectionist networks: An experiment in speech recognition. *Technical Report MS-CIS-86-78*, University of Pennsylvania.

Appendix

The appendix enumerates some R scripts that were used to create illustrations in section 3. Scripts for other figures are not listed here. For reference, Fig. 1-3 were created by Praat, Fig. 4-6 were created by R.

Fig. 8 Gaussian distribution

```
x <- seq(-4,4,0.01)
x_i = 0.7

par(mar=c(2.2,2.2, 0.5, 0.5))
plot(x, dnorm(x), type="l", xlab="", ylab="", lwd=3)
segments(x_i, -1, x_i, dnorm(x_i)+0.02, lwd=1)
segments(-5, dnorm(x_i), (x_i+0.1), dnorm(x_i), lwd=1)

mtext(side=1, at=x_i, cex=1.4, paste("x=", x_i, sep=""), line=1)
text(  x_i, dnorm(x_i), pos=4, cex=1.4, labels=paste("p=", sprintf("%0.3f", dnorm(x_i)), sep="") )
```

Fig. 9 Two random variables with no significant correlation

```
n = 1*1000
x1 <- rnorm(n)
x2 <- rnorm(n)

lim = 5; par(mar=c(2.2,2.2, 0.5, 0.5))
plot(x1, x2, pch=20, ylim=c(-lim,lim), xlim=c(-lim,lim))

abline(v=0, h=0, col="darkgrey")
text(-lim, lim*1.0, cex=1.4, pos=4, labels=expression(paste(mu[1], "=", 0, ", ", sigma[1], "=", 1)))
text(-lim, lim*0.9, cex=1.4, pos=4, labels=expression(paste(mu[2], "=", 0, ", ", sigma[2], "=", 1)))
text(-lim,  lim*0.8, cex=1.4, pos=4, labels=expression(paste(rho%~~%0)))
```

Fig. 10 Two random variables with strong correlation

```
library("mvtnorm")
n = 1000
lim = 5

x1.sd = 1.0
x2.sd = 1.0
rho=0.8
sigma = matrix(c(x1.sd^2, rho*x1.sd*x2.sd, rho*x1.sd*x2.sd, x2.sd^2), nrow=2)
x12 = rmvnorm(n, c(0,0), sigma=sigma)

# recalculate values
x1.mu = mean(x12[,1]);  x1.sd = sd(x12[,1])
x2.mu = mean(x12[,2]);  x2.sd = sd(x12[,2])
rho = cor(x12)[1,2];    rho.lab = sprintf("%0.3f", rho)
x12.sigma = matrix(c(x1.sd^2, rho*x1.sd*x2.sd, rho*x1.sd*x2.sd, x2.sd^2), nrow=2)

lim = 5; par(mar=c(2.2,2.2, 0.5, 0.5))
plot(x12, pch=20, ylim=c(-lim,lim), xlim=c(-lim,lim))

abline(0, rho, col="gray")
text(-lim, lim*1.0, cex=1.4, pos=4, labels=expression(paste(mu[1], "=", 0, ", ", sigma[1], "=",  1)))
text(-lim, lim*0.9, cex=1.4, pos=4, labels=expression(paste(mu[2], "=", 0, ", ", sigma[2], "=",  1)))
text(-lim, lim*0.8, cex=1.4, pos=4, labels=bquote(rho~"="~.(rho.lab)))
```

Fig. 11 Probability density for two-dimensional observations

```
library("mvtnorm")
lim = 5
x = y = seq(-lim, lim, length=60)

x1.mu =0;  x1.sd = 1;   rho=0.8
x2.mu =0;  x2.sd = 0.5
x12.sigma = matrix(c(x1.sd^2, rho*x1.sd*x2.sd, rho*x1.sd*x2.sd, x2.sd^2), nrow=2)

f1 = function(x,y){dmvnorm(c(x,y), mean=c(x1.mu,x2.mu), sigma=x12.sigma )}
z = outer(x, y, function(x,y)mapply(f1,x,y)) # For non-vectorized 'f1'

par(  cex.lab=1.6, mar=c(0, 0.0 ,0, 0)  )
persp(x,y,z,theta = 80, phi = 30, zlab="probability", expand=0.6, r=4, shade=0.1) -> res
marg.y = 0.9*(max(z) * colSums(z) / max(colSums(z)))
marg.x = 0.9*(max(z) * rowSums(z) / max(rowSums(z)))

lines (trans3d(x=-lim, y,      z = marg.y, pmat = res), col = "gray")
lines (trans3d(x,      y=lim,  z = marg.x, pmat = res), col = "gray")
```

Fig. 12 A multimodal distribution with 3 Gaussian components

```
library("mvtnorm")

lim = 5

x <- seq(-lim,lim,0.01)
mu1=-2; sd1=0.5
mu2=0;  sd2=1
mu3=3;  sd3=0.3
y1 <- dnorm(x, mean=mu1, sd=sd1)
y2 <- dnorm(x, mean=mu2, sd=sd2)
y3 <- dnorm(x, mean=mu3, sd=sd3)
w1=1.0
w2=0.8
w3=1.2
Y = w1*y1+w2*y2+w3*y3

plot(x,Y, type="l", lwd=3)

lines(x, w1*y1, col="gray")
lines(x, w2*y2, col="gray")
lines(x, w3*y3, col="gray")
lines(x, Y, col="black")

segments(mu1, 0, mu1, w1*dnorm(mu1, mean=mu1, sd=sd1), lwd=1, col="gray")
segments(mu2, 0, mu2, w2*dnorm(mu2, mean=mu2, sd=sd2), lwd=1, col="gray")
segments(mu3, 0, mu3, w3*dnorm(mu3, mean=mu3, sd=sd3), lwd=1, col="gray")
```

Fig. 13 A Gaussian mixture model with bimodal bivariate distribution

```
library("mvtnorm")

lim = 5
x = y = seq(-lim, lim, length=50)

mu1=2;  sd1=1.5;
mu2=-2; sd2=1.2;
rho12=-0.4
sigma12=matrix(c(sd1^2, rho12*sd1*sd2, rho12*sd1*sd2, sd2^2), nrow=2)
f12 = function(x,y){dmvnorm(c(x,y), mean=c(mu1,mu2), sigma=sigma12)}
z12 = outer(x, y, function(x,y)mapply(f12,x,y)) # For non-vectorized 'f1'

mu3=-1.3; sd3=1;
mu4=2;   sd4=0.85;
rho34=0.65
sigma34=matrix(c(sd3^2, rho34*sd3*sd4, rho34*sd3*sd4, sd4^2), nrow=2)
f34 = function(x,y){dmvnorm(c(x,y), mean=c(mu3,mu4), sigma=sigma34)}
z34 = outer(x, y, function(x,y)mapply(f34,x,y)) # For non-vectorized 'f1'

Z = z12 + z34

par(
  cex.lab=1.6,
  mar=c(0, 0.0 ,0, 0)
)
persp(x,y,Z,theta = 80, phi = 30, zlab="probability", expand=0.6, r=4, shade=0.1) -> res

marg.y = 0.9*(max(Z) * colSums(Z) / max(colSums(Z)))
marg.x = 0.9*(max(Z) * rowSums(Z) / max(rowSums(Z)))

lines (trans3d(x=-lim, y[],  z = marg.y, pmat = res), col = "gray")
lines (trans3d(x,      y=lim,z = marg.x, pmat = res), col = "gray")
```