KTH Speech, Music
and Hearing

Cluster analysis methods for speech recognition

Julien Neel (ENST Paris)

Supervisor: Giampiero Salvi

Approved: 2005-02-18

Examiner: Rolf Carlson

.....
(signature)

Stockholm
17th February 2005

Examenarbete i Talteknologi
(Master Thesis in Speech Technology)
Department of Speech, Music and Hearing
Royal Institute of Technology
S-100 44 Stockholm

KTH Tal, musik och hörsel

Examensarbete i Talteknologi

Klustringmetoder för taligenkänning

Julien Neel (ENST Paris)

Godkänt:	Examinator:	Handledare:
2005-02-18	Rolf Carlson	Giampiero Salvi

Abstract

How well can clustering methods capture a phonetic classification? Can the "appropriate" number of clusters be determined automatically? Which kinds of phonetical features group together naturally? How can clustering quality be measured?" To what extent is an automatic clustering method reliable in this case?

This study tries to answer these questions with a number of experiments conducted on speech data using K-means and fuzzy K-means clustering. Optimal number of clusters were determined with the Davies Bouldin and I family indexes.

For this report, the data considered was extracted from the `TIMIT` database, a corpus of read speech with phoneme transcription. A small clustering toolbox for Matlab was implemented. It computed clusterings using various classical methods and cluster validity indexes to assess quality. A number of benchmark tests were run on the `IRIS` data as well as synthetic data.

The speech examples, show that when clustering phonemes, certain acoustical and articulatory features can be captured. Fuzzy clustering can improve cluster quality. The indexes, which yield an optimal number of clusters according to a criterion to optimize, do not give us the number of phonemes, but can help spot broad phonetic classes. Another phonetic classification, based on both acoustic and articulatory features is more natural than broad phonetic classes.

Clustering methods can also help ranking the importance of acoustical/articulatory features such as those of Jackobson, Halle and Fant. Some of these features emerge naturally from the observations.

Acknowledgments

I would like to thank Björn Granström and Rolf Carlson for their warm welcome and for giving me the opportunity to do my internship as a master thesis student in the Speech Department of KTH, Giampiero Salvi for tutoring me in discovering research, my fellow colleagues Linda Oppelstrup for the late nights, Elina Eriksson and Sara Öhgren for enlightening days, and Anders Lundgren for his curiosity, Gunilla Svanfeldt and Preben Wik for the advice and chats, Niclas Horney for having fixed my computer and unlocked my account every two weeks, and Catrin Dunger for having been so helpful. Thanks to all the members of the Speech Department for the great working atmosphere and the wonderful Christmas party.

List of Figures

List of Tables

List of Abbreviations

ASR	Automatic speech recognition
DB	Davies Bouldin index
EM	Expectation maximization algorithm
$\mathbf{F_{meas}}$	F-measure index
GMM	Gaussian mixture models
MACF	Maximum of the autocorrelation function
MFCC	Mel frequency cepstrum coefficients
PCA	Principal component analysis
SVC	Support vector clustering
ZCR	Average zero-crossing rate

Contents

1 Introduction

1.1 Method and problem formulation

Clustering methods have traditionally been used in order to find emerging patterns from data sets with unknown properties. Clustering is a ill-defined problem for which there exist numerous methods (see [4, 10, 16, 1]). Unfortunately, many articles that dealt with clustering did not take into account high dimensional data sets, as it is the case in speech.

In this study a number of clustering algorithms, including K-means and fuzzy K-means, have been tested both on benchmark data (`IRIS` and various synthetic data clouds with ellipsoidal or chainlike shapes, such as rings) and on the `TIMIT` speech database, with recordings of American English speakers. Benchmark data has been also tested with a Support Vector Clustering procedure, but for reasons of computational time on simple data sets, this method was not retained for speech data.

The `Matlab` routines extracted feature vectors from the feature files, classified them using the `TIMIT` transcription file, stored the results and displayed various visualizations for interpretation. Various data sets for training and testing were prepared. The focus of the experiments was initially to estimate the performance of clustering methods in a classification task where each cluster was assigned the phonetic class closest to it.

Other experiments were conducted in order to investigate upon the information such clustering procedures could find. Clustering results were analyzed visually in 2D, as histograms resuming how clusters captured the classes, and with `Wavesurfer`, in order to display cluster assignments in time. Since the data contained overlapping clusters and outliers and had many dimensions, it was difficult to estimate visually in two dimensions how well natural groups appeared.

Therefore, additional indexes and validity indexes to assess cluster quality were computed: the F-measure, purity and the partition coefficient PC. These provide several measures of the quality of the clustering with respect to the classification.

1.2 Outline of the report

The report is organized as follows: Section 2 presents basics on phonetic feature models and on the clustering procedures used. Cluster validity is also discussed here. Section 3 describes the experimental settings and method. Some results are presented in Section 4 and discussed in section 5. Finally, section 6 concludes the report.

2 Theory

2.1 Phonetic feature model

2.1.1 Phonemes

In linguistics, the fundamental unit of speech is the phoneme. Phonemes are the basic theoretical units for describing how speech conveys meaning. They compose a family of similar sound groups. Its members, the allophones, are considered equivalent for a given language.

In English for example, [p^h] and [p] are allophones of the phoneme p, as in *pit* and *spit*. The first consonant of *pit* has an extra puff of air after it which is not found after the [p] of *spit*. However, switching allophones of the same phoneme won't change the meaning of the word: [sp^{h}It] still means *spit*. Switching allophones of different phonemes will change the meaning of the word or result in a nonsense word: [skIt] and [stIt] do not mean *spit*.

Different languages can have different groupings for their phonemes. [p] and [p^h] belong to the same phoneme in English, but to different phonemes in Chinese. In Chinese, switching [p] and [p^h] does change the meaning of the word.

A broad transcription uses only one symbol for all allophones of the same phoneme. This is enough information to distinguish a word from other words of the language. What details you have to include in a broad transcription will depend on what language or dialect you are transcribing.

In English, there are about 42 phonemes. They tradionally fall into the following categories: vowels (front, mid or back), diphthongs, semi-vowels (glides and liquids) and consonants (stops, plosives, nasals, fricatives and affricates).

Phonetic classes	Sample phonemes
stops	b, d, g, p, t, k
affricates	ch
fricatives	s, sh, z, f
nasals	m, n, ng
semivowels & glides	l, r, w, y
vowels	ah, aa, eh, iy, uw

Table 1: Broad phonetic classes in speech

Each sound has particular spectral characteristics. These characteristics change continuously in time. The patterns of change give cues to phone identity. However, speech spectra (see figure 1) also include numerous sources of variability. Emotion and phonetic context for example can change a person's way of speaking, and it can be a great source of speaker intravariability. Spectra vary even more when the same utterance is pronounced by speakers of different gender, speaking style, etc.

2.1.2 Feature vectors for speech

The underlying assumption of local stationarity in most speech processing methods is that the properties of the speech signal change relatively slowly with time. This assumption leads to a variety of short-time processing methods in which short segments of the speech

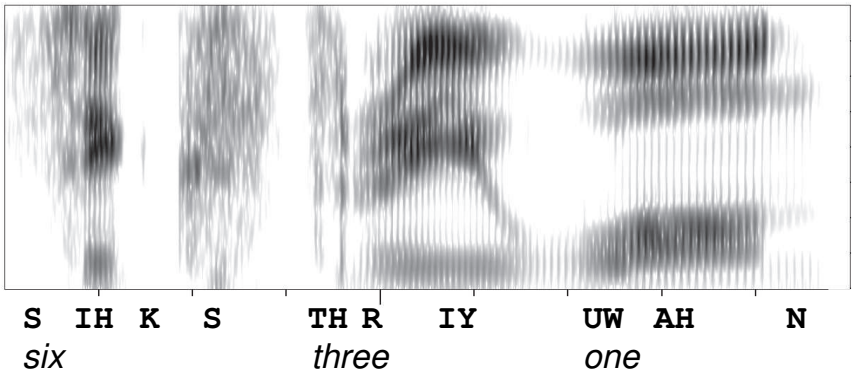

Figure 1: Speech spectral characteristics

signal are isolated and processed as if they were short segments from a sustained sound with fixed properties. These short segments called time analysis frames overlap one another. The result of the processing on each frame is a new time-dependant sequence which can serve as a representation of the speech signal.

The feature vectors represent the speech signal. Each dimension requires processing methods that involve the waveform of speech signal directly and others require using the spectrum of the speech signal. They are called time-domain and frequency-domain methods. Measurements such as energy, zero-crossing rate and the autocorrelation function are of the first type, and mel-cepstrum coefficients are of the second type.

The feature dimensions are described below.

Short-time energy: The amplitude of the speech signal varies appreciably with time. In particular, the amplitude of unvoiced segments is generally much lower than the amplitude of voiced segments. The short-time energy of the speech signal provides a convenient representation that reflects these amplitude variations. The energy function can also be used to locate approximately the time at which voiced speech becomes unvoiced, and vice versa, and, for very high quality speech (high signal-to-noise ratio), the energy can be used to distinguish speech from silence. The formula for short-time energy is:

$$E_n = \sum_{-M \leq m \leq M} [x(m) * w(n-m)]^2 \tag{1}$$

An illustration of short-time energy can be found in figure 2.

Short-time average zero-crossing rate: In the context of discrete-time signals, a zero-crossing is said to occur if successive samples have different algebraic signs. This measure gives a reasonable way to estimate the frequency of a sine wave. Speech signals are broadband signals and the interpretation of average zero-crossing is therefore much less precise. Since high frequencies imply high zero-crossing rate and low frequencies imply low zero-crossing rates, there is a strong correlation between zero crossing rate and energy distribution with frequency. A reasonable generalization is that if the zero-crossing rate is high, the speech signal is unvoiced, while if the zero-crossing rate is low, the speech signal

is voiced. This however, is a very imprecise statement because high and low remain to be defined. Energy and zero-crossing rate can help to discriminate speech from silence, making it possible to process only the parts of the input that correspond to speech. The formula for short-time average zero-crossing rate is:

$$Z_n = \sum_{-M \leq m \leq M} |\, sgn[x(m)] - sgn[x(m-1)]\,| * w(n-m) \tag{2}$$

An illustration of short-time average zero-crossing rate can be found in figure 2.

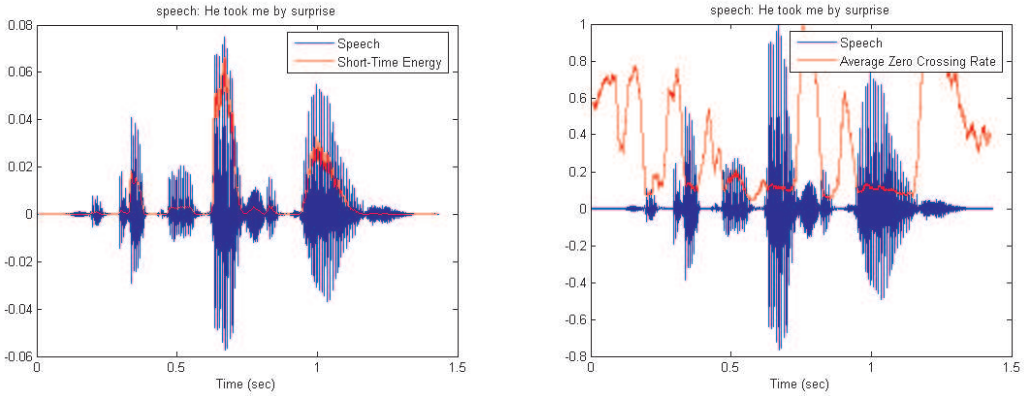

Figure 2: Illustration of the short-time energy and ZCR coefficients

Short-time autocorrelation function: It is a version of the autocorrelation function adapted to the short-time representation. It is an even function, attains its maximal value in 0, and this supremum is equal to the energy for deterministic signals or the average power for random or periodic signals. Periodicity in the signal is given by the first maximum in the function. This property makes the autocorrelation function a means for estimating periodicity in many signals, including speech. It contains much more information about the detailed structure of the signal. The formula for short-time autocorrelation is:

$$R_n(k) = \sum_{m=0}^{N-1-k} [x(n+m)w'(m)] * [x(n+m+k)w'(m+k)] \tag{3}$$

Mel-cepstrum coefficients: Front-end analysis aims at extracting phonetically significant acoustic information for the human ear. This useful information can be captured using the short-term spectrum and mathematical approximations concerning its local stationarity. Human ability to perceive a certain frequency f is influenced by the energy in a critical band of frequencies, whose size depends on the considered frequency f. Because the filtering operation of the vocal tract is the most influential factor in determining phonetic properties of speech sounds, it is desirable to separate out the excitation component $g(t)$ from the filter component $h(t)$ in the signal $s(t)$:

$$s(t) = g(t) * h(t) \tag{4}$$

Cepstral analysis is a technique for estimating a separation of the source and filter components. The mel-cepstrum, introduced by Davies and Mermelstein, exploits auditory principles, as well as the decorrelating property of the cepstrum. In addition, the mel-cepstrum is amenable to compensation for convolutional channel distortion. As such, the MFCCs have proven to be one of the most successful feature representations in speech-related recognition tasks.

A filter bank, with triangular filters placed according to an empirical log-linear scale called the mel-scale, is constructed to resemble the human auditory system with more channels at low frequencies and fewer at high. This makes it easier to estimate the frequencies of the formants and thus the phone being uttered. Filtering the spectrum (also called *liftering*) can be applied to remove certain components or alter the relative influence of the different components. Most of the influence of the fundamental frequency is actually removed from the spectrum. The resulting spectrum can be much smoother than the original and show the formant peaks more clearly (see [12]).

2.1.3 Problems linked to phonetic representation

Although well-defined gaps are often perceived between words in a language we speak, this is often an illusion. Examining the actual the actual physical speech signal, one often finds undiminished sound energy at word boundaries. Indeed, Indeed, a drop in speech energy is as likely to occur within a word as between words. This property of speech is particularly compelling when we listen to someone speaking a foreign language. The speech appears to be a continuous stream of sounds with no obvious word boundaries. It is our familiarity with our own language that leads to the illusion of word boundaries.

Within a single word, even greater segmentation problems exist. These intra-word problems involve the identification of the basic vocabulary of speech sounds: phonemes. A segmentation problem arises when the phonemes composing a spoken word need to be identified. Segmentation at this level is like recognizing a manually written text where one letter runs into another. Also, as in the case of writing, different speakers vary in the way they produce the same phonemes. The variation among speakers is strong.

In this report, we will be dealing with general segmenting of speech into words/phonemes. We will consider each frame as a data point, therefore the utterance of a given phoneme will be represented as many times as there are such data points.

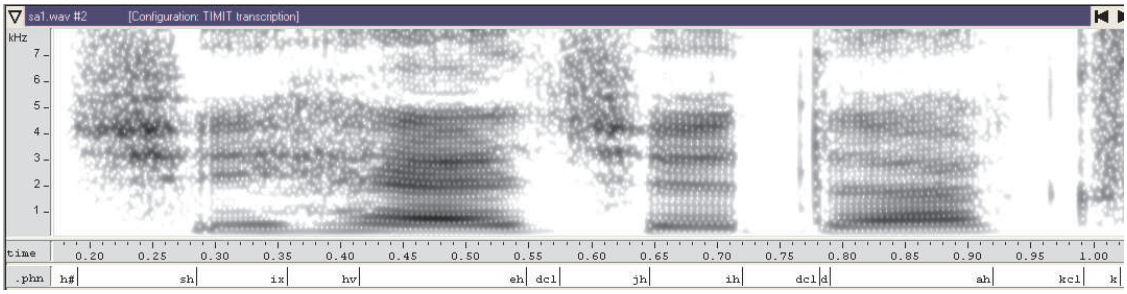

Figure 3: A sample spectrum from the `TIMIT` speech data base

A further difficulty in speech perception involves a phenomenon known as co-articulation. Consider in figure 3) the spectrum of a female speaker in the New England dialect uttering

"she had your dark" and let's focus on the word *dark*. As the vocal tract is producing one sound, say the stop `[d]` in the phoneme sequence `[d, ah, kcl, k]`, it is moving towards the shape it needs for the vowel `[ah]`. Then, as it is saying the `[ah]`, it is moving to produce the voiced stop `[k]`, with the closure interval of the stop `kcl` in between. In effect, the various phenomenons overlap. This means additional difficulties in segmenting phonemes, and it also means that the actual sound produced for one phoneme will be determined by the context of the other surrounding phonemes.

2.1.4 Phoneme classifications

Classical Phonetics use the place-manner classification system for consonants and the high-low / front-back system for vowels. The main purpose is to enable the phonetician to specify how particular sounds are produced with respect to their articulation. It becomes possible to use the features or parameters of the classification system to label whole sets of sounds or articulations. Thus we might refer to the set of all plosives or the set of all voiced plosives (three in English), or the set of all voiced alveolar plosives (one only in English) - and so on, cutting horizontally and vertically around the consonant matrix. Similarly, for vowels, the set of all front vowels, or the set of all rounded vowels, and so on.

Distinctive feature theory was first formalized by Jackobson ad Halle in 1941 (see [7]), and remains one of the most significant contributions to phonology. There have been numerous refinements to the set since that date. These features are binary, which means that a phoneme either has the feature (e.g.: `[+voice]`) or doesn't have the feature (`[-voice]`). A small set of features is able to differentiate between the phonemes of any single language. Distinctive features may be defined in terms of articulatory or acoustic features: for example, `[grave]` is an acoustic feature (concentration of low frequency energy) and `[high]` is an articulatory feature related to tongue height, but it can also be readily defined in acoustic terms. Jackobson's features are primarily based on acoustic descriptions.

+/−	Acoustic (+)	Articulatory (+)
grave/acute	concentration of energy in lower frequencies	peripheral
voiced/voiceless	periodic low frequency excitation	periodic vibration of vocal chords
continuant/discontinuous	no abrupt transition sound/silence	no turning on/off of source
strident/mellow	higher intensity noise	rough-edged

Table 2: Illustration of 4 of the 14 binary features from Jackobson's classification

2.2 Clustering basics

2.2.1 Introduction

Clustering is the process of grouping together similar objects. The resulting groups are called clusters. Clustering algorithms group data points according to various criteria. Unlike most classification methods, clustering handles data that has no labels, or ignores

the labels while clustering. The concept mostly utilizes geometric principles, where the samples are interpreted as points in a d-dimensional Euclidian space, and clustering is made according to the distances between points: usually, points which are close to each other will be allocated to the same cluster.

This method is an unsupervised process since there are no predefined classes and no examples that would indicate grouping properties in the data set. The various clustering algorithms are based on some assumptions in order to define a partitioning of a data set. Most methods behave differently depending on the features of the data set and the initial assumptions for defining groups. Therefore, in most applications, the resulting clustering scheme requires some sort of evaluation regarding its validity. Evaluating and assessing the results of a clustering algorithm is the main object of cluster validity ([5, 6]).

2.2.2 Applications

Clustering is one of the most useful method in the data mining process for discovering groups and identifying interesting distributions and patterns in the underlying data. Thus, the main concern in the clustering process is to reveal the organization of patterns into sensible groups, which allow us to discover similarities and differences, as well as to derive useful inferences about them. Clustering makes it possible to look at properties of whole clusters instead of individual objects - a simplification that might be useful when handling large amounts of data. It has the potential to identify unknown classification schemes that highlight relations and differences between objects.

2.2.3 Problems

In the literature (see [11, 10, 16, 2, 4]), a wide variety of algorithms have been proposed for different applications and sizes of data sets, but there seems to be no unifying theory of clustering. Most clustering benchmarks deal with low dimensional real world or synthetic data sets, unlike speech data.

A frequent problem many clustering algorithms encounter is the choice of the number of clusters. Quite different kinds of clusters may emerge when its value is changed. Most clustering methods require the user to specify a value, though some provide means to estimate the number of clusters inherent within the data. Among those, Support Vector Machines, model-based methods that use the Bayes information criterion for model selection, or different kinds of indexes, such as Davies-Bouldin's index.

Many clustering algorithms prefer certain cluster shapes and sizes, and these algorithms will always assign the data to clusters of such shapes even when there were no such clusters in the data. For example, K-means is only able to form spherical clusters, and if applied to concentric ring shaped data, locates all the centroids together in the center of the rings

Handling strongly overlapping clusters or noise in the data is often difficult. For instance, when clusters are not well separated and compact, the distance between clusters as the average distance between pairs of samples does not work well. Some algorithms get stuck at local minima and the result is largely dependent on initialization, one of the major drawbacks of the K-means algorithm. Large data sets or vectors with many components can be a serious computational burden.

2.2.4 Clustering algorithms overview

A multitude of clustering methods are proposed in the literature (see [11, 10, 16, 2, 4]). Clustering algorithms can be classified according to the type of data input to the algorithm, the clustering criterion defining the similarity between data points, and the theory and fundamental concepts on which clustering analysis techniques are based (e.g.: fuzzy theory, statistics, etc.). Thus according to the method adopted to define clusters, the algorithms can be broadly classified into the following types:

Partitional clustering attempts to directly decompose the data set into a set of disjoint clusters. More specifically, they attempt to determine an integer number of partitions that optimize a certain criterion function. The criterion function may emphasize the local or global structure of the data and its optimization is an iterative procedure.

Hierarchical clustering proceeds successively by either merging smaller clusters into larger ones, or by splitting larger clusters. The result of the algorithm is a dendrogram (tree) of clusters which shows how the clusters are related. By cutting the dendrogram at a desired level, a clustering of the data items into disjoint groups is obtained. These methods are often not suitable for large data sets because of computational time.

Density-based clustering groups neighboring objects into clusters based on density conditions. It is model based and parametric. Gaussian Mixture Models (or weighted sum of gaussian distributions) is a typical example.

Grid-based clustering is mainly proposed for spatial data mining. Their main characteristic is that they quantize the space into a finite number of cells and they do all operations on the quantized space. Self-Organizing Feature Maps are an example of such methods.

Support Vector Clustering maps by means of a gaussian kernel to a high dimensional feature space, where the algorithm seeks a minimal enclosing sphere. This sphere, when mapped back to the original data space, can separate into several components, each enclosing a separate cluster of points. Data points are either strictly inside clusters or become either Support Vectors, which lie on cluster boundaries, or Bounded Support Vectors, which lie outside the boundaries. Soft constraints are incorporated by adding slack variables which help coping with outliers and overlapping clusters. The lower the number of Support Vectors, the smoother the boundaries. For high dimensional data sets, the number or Support Vectors jumps, nearly all points being Support Vectors (see [2]).

2.3 Clustering algorithms used

2.3.1 K-means

K-means clustering is an elementary but very popular approximate method that can be used to simplify and accelerate convergence. Its goal is to find K mean vectors $(\mu_1, \ldots, \mu_K)$ which will be the K cluster centroids. It is traditional to let K samples randomly chosen from the data set serve as initial cluster centers.

The algorithm is then:

- Step 0: set the number of clusters K
- Step 1: initialize cluster centroids $(\mu_1, \ldots, \mu_K)$
- Step 2: classify the samples according to the nearest μ_k
- Step 3: recompute $(\mu_1, \ldots, \mu_K)$ until there is no significant change.

This method is illustrated in figure 4. At a given iteration (top left), points are distributed in two clusters. Their centroids are computed (top right). Then data points are reallocated to the cluster whose centroid is nearest (bottom right) and the new cluster centroids are computed (bottom left).

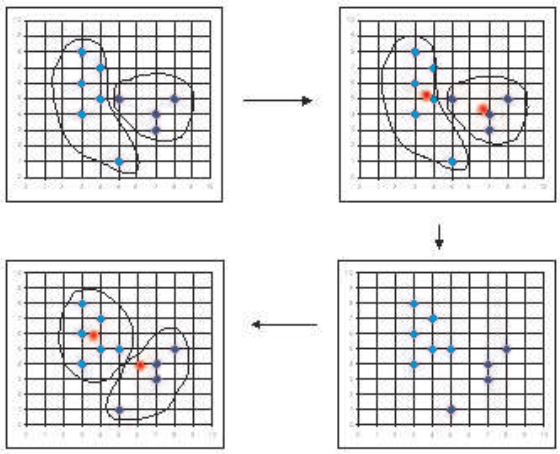

Figure 4: Illustration of the K-means algorithm

The computational complexity of the algorithm is $O(NdKT)$, where N is the number of samples, d the dimensionality of the data set, K the number of clusters and T the number of iterations.

In general, *K-means* does not achieve a global minimum over the assignments. In fact, since the algorithm uses discrete assignment rather than a set of continuous parameters, the minimum it reaches cannot even be properly called a local minimum. In addition, results can greatly depend on initialization. Non-globular clusters which have a chain-like shape are not detected with this method. Despite these limitations, the algorithm is used fairly frequently as a result of its ease of implementation (see [3])

2.3.2 Fuzzy K-means

In every iteration of the classical *K-means* procedure, each data point is assumed to be in exactly one cluster. This condition can be relaxed by assuming each sample x_i has some graded or fuzzy membership in a cluster ω_k. These memberships correspond to the probabilities $p(\omega_k|x_i)$ that a given sample x_i belongs to class ω_k.

The *fuzzy K-means* clustering algorithm seeks a minimum of a heuristic global cost function:

$$J_{fuz} = \sum_{k=1}^{K} \sum_{i=1}^{N} [p(\omega_k|x_i)]^b \, d_{ik} \tag{5}$$

where: $d_{ik} = \| x_i - \mu_k \|^2$ and b is a free parameter chosen to adjust the blending of different clusters. If $b = 0$, J_{fuz} is merely a sum-of-squares criterion with each pattern assigned to a single cluster. For $b > 1$, the criterion allows each pattern to belong to multiple clusters.

The solution minimizes the fuzzy criterion. In other words one computes the following for each $1 \leq k \leq K$:

$$p(\omega_k|x_i) = \frac{(d_{ik})^{(\frac{1}{1-b})}}{\sum_{l=1}^{K} (d_{il})^{(\frac{1}{1-b})}} \tag{6}$$

and for each dimension $1 \leq j \leq d$:

$$\mu_k(j) = \frac{\sum_{i=1}^{N} x_i(j) \, [p(\omega_k|x_i)]^b}{\sum_{i=1}^{N} [p(\omega_k|x_i)]^b} \tag{7}$$

The cluster means and probabilities are estimated iteratively according to the following algorithm:

- Step 0: set the number of clusters K and parameter b
- Step 1: initialize cluster centroids $(\mu_1, \ldots, \mu_K)$ and $(p(\omega_k|x_i))_{ik}$
- Step 2: recompute centroids $(\mu_1, \ldots, \mu_K)$ and $(p(\omega_k|x_i))_{ik}$ until there is no significant change.

The incorporation of probabilities as graded memberships sometimes improves the convergence of *K-means* over its classical counterpart. One drawback of the method is that the probability of membership of a point in a cluster depends implicitly on the number of clusters, and when this number is specified incorrectly, results can be misleading (see [8]).

2.3.3 Gaussian Mixture Models and EM algorithm

Model-based clustering assumes that the data are generated by a mixture of probability distributions in which each component represents a different cluster. The number of desired clusters has to be specified prior to the clustering procedure.

Consider a set of N points $(x_1, \ldots, x_N)$ in $\mathbb{R}^d$ to be clustered into K groups. The data is seen as N observations of a d-dimensional random vector with density:

$$\phi_\theta(x) = \sum_{k=1}^{K} \alpha_k \phi_k(x) \tag{8}$$

where ϕ_k is the density associated to the normal distribution $N(\mu_k, \Sigma_k)$ and α_k the weight of this component in the mixture ($\sum \alpha_k = 1$). In order to simplify computation, we will suppose the covariance matrices Σ_k are diagonal and denote σ_k the vector of the diagonal elements. This means that we assume that the clusters have spherical shapes with

variable volumes. The parameter of the model is $\theta = (\alpha, \mu, \sigma)$, where $\alpha = (\alpha_1, \ldots, \alpha_K)$, $\mu = (\mu_1, \ldots, \mu_K)$ and $\sigma = (\sigma_1, \ldots, \sigma_K)$.

The *Expectation-Maximization* or *EM* algorithm gives a mean to estimate the parameters of the model. Let $\phi_{\mu,\sigma}$ denote the density of a d-dimensional random gaussian vector associated to the diagonal elements of the covariance matrix and to the vector of means μ. Denote $p_\theta(\omega_k|x_i)$ the posterior probability that, given x_i, the point belongs to cluster ω_k:

$$p_\theta(\omega_k|x_i) = \frac{\alpha_k \phi_{\mu_k,\sigma_k}(x_i)}{\sum_{l=1}^{K} \alpha_l \phi_{\mu_l,\sigma_l}(x_i)} \tag{9}$$

From one estimation of the parameter $\theta = (\alpha, \mu, \sigma)$ to the next $\theta' = (\alpha', \mu', \sigma')$, one computes the log-likelihood of the model. Given data x and a model parameterized by θ, we seek a θ' that maximizes the likelihood of x. In other words, one computes for each $1 \leq k \leq K$:

$$\alpha'_k = \frac{1}{N} \sum_{i=1}^{N} p_\theta(\omega_k|x_i) \tag{10}$$

and for each dimension $1 \leq j \leq d$:

$$\begin{aligned} \mu'_k(j) &= \frac{\sum_{i=1}^{N} x_i(j) p_\theta(\omega_k|x_i)}{\sum_{i=1}^{N} p_\theta(\omega_k|x_i)} \\ \sigma'_k(j) &= \sqrt{\frac{\sum_{i=1}^{N} p_\theta(\omega_k|x_i)(x_i(j) - \mu'_k(j))^2}{\sum_{i=1}^{N} p_\theta(\omega_k|x_i)}} \end{aligned} \tag{11}$$

One proves log-likelihood is an increasing function of θ. One considers that the optimal value for θ is found after a certain number of iterations or when the log-likelihood function stabilizes.

2.4 Cluster indexes and validity measures

2.4.1 Davies-Bouldin index

A similarity measure $R(K_i, K_j) = R_{ij}$ between two clusters K_i and K_j is defined. It is based on a measure of dispersion $s(K_i) = s_i$ of a cluster K_i, and a dissimilarity measure $d(K_i, K_j) = d_{ij}$ between two clusters K_i and K_j. R_{ij} is defined to be non negative and symmetric.

The Davies-Bouldin or DB index for K clusters is defined as follows:

$$DB = \frac{1}{K} \sum_{j=1}^{K} \max_{\substack{1 \leq i \leq K \\ i \neq j}} R_{ij} \tag{12}$$

where:

$$R_{ij} = \frac{s_i + s_j}{d_{ij}} \tag{13}$$

and, if z_i denotes the centroid of cluster K_i, with:

$$\begin{aligned} s_i &= \sqrt{\frac{1}{card(K_i)} \sum_{x \in K_i} \|x - z_i\|^2} \\ d_{ij} &= \max_{1 \leq i,j \leq K} \|z_i - z_j\| \end{aligned} \tag{14}$$

The DB index is the average similarity between each cluster and its most similar one. It is desirable for the clusters to have the least possible similarity to each other, so the smaller the DB index, the more clusters tend to be compact and not overlap, thus better expected separation. The number of clusters which minimizes the DB index is the optimal one. It is generally admitted that this index exhibits no trends with respect to the number of clusters.

2.4.2 The I indexes

Consider a data set of N points partitioned into K clusters. The I index (see [9]) is defined as follows:

$$I = \left(\frac{D_K}{K \times E_K} \right)^p \tag{15}$$

where K is the number of clusters, N the number of points and E_K is:

$$\begin{aligned} E_K &= \sum_{\substack{1 \leq i \leq N \\ 1 \leq k \leq K}} u_{ki} \, \|x_i - z_k\| \\ D_K &= \max_{1 \leq i,j \leq K} \|z_i - z_j\| \end{aligned} \tag{16}$$

where $[u_{ij}]_{\substack{1 \leq i \leq N \\ 1 \leq j \leq K}}$ is a partition matrix for the data such that: $u_{ki} = 1$ if x_i is in cluster K_j of centroid z_j. The value of K for which this index is maximized is considered to be the correct number of clusters.

Each of the factors composing the index penalize it in the following way: The first factor D_K, which measures the maximal separation between two clusters over all possible pairs of clusters, will increase with the number K, hence reducing the index. The second factor $\frac{1}{K}$ will try to reduce the index as the number of cluster increases. The third factor $\frac{1}{E_K}$, which measures the total fuzzy dispersion, will penalize the index as it is increased. The power p is used to control the contrast between the different cluster configurations. For the present report, $p = 2$.

The I_{fuz} index is defined the same way as the I index, but with the different cluster membership values u_{ij} ranging in the entire interval $[0, 1]$, instead of taking values in $\{0, 1\}$.

2.4.3 F-measure

The F-measure F_{meas} is an index which describes how well a clustering configuration fits a classification. It also gives a means to compare different clusterings and determine which is most likely to correspond to the classification.

Purity of a clustering describes the average purity of the clusters obtained. In other words, it is a measure of how good the clustering is, if one seeks to have clusters which represent only one class.

Consider a data set containing C classes and partitioned into K clusters. Precision and

recall give a means to compare each cluster K_i with each class C_j. They are defined as:

$$\begin{aligned} p(K_i, C_j) &= \frac{card(K_i \cap C_j)}{card(K_i)} \\ r(K_i, C_j) &= \frac{card(K_i \cap C_j)}{card(C_j)} \end{aligned} \tag{17}$$

Improving recall and improving precision are antagonistic goals and efforts to improve one often result in degrading the other. High precision means that most elements in the cluster are from the same class, whereas high recall is synonymous with most elements from the class were grouped the cluster.

The F-measure for a cluster K_i and a class C_j is defined as the harmonic average of precision and recall:

$$F(K_i, C_j) = F_{ij} = \frac{1}{\frac{1}{p_{ij}} + \frac{1}{r_{ij}}} \tag{18}$$

Then one defines the F-measure of cluster K_i over all the classes as:

$$F(K_i) = F_i = \max_{1 \leq j \leq C} F_{ij}$$

and the total F-measure for the entire clustering as the weighted average of the F_i over all clusters:

$$F_{meas} = \sum_{1 \leq i \leq K} \frac{card(K_i)}{N} F_i \tag{19}$$

The closer the F_{meas} is to 1, the more the clustering fits the classification.

2.4.4 Purity

In a similar way, one can define the purity of a clustering ρ_i of each cluster K_i as the the highest precision p_{ij} reached over the different classes:

$$\rho_i = \max_{1 \leq j \leq C} p_{ij} \tag{20}$$

The weighted average of ρ_i over all clusters yields a measure of quality of the whole clustering:

$$\rho = \sum_{1 \leq i \leq K} \frac{card(K_i)}{N} \rho_i \tag{21}$$

The closer purity is to 1, the more clustering tends to break down classes into clusters one by one. In other words, we try to perform a covering of each class using clusters, in order for clusters to be able to define classes in return.

2.4.5 Principal component analysis

Principal component analysis, or PCA, helps to discover or to reduce the dimensionality of the data set and to identify new meaningful underlying variables. It performs a linear transformation on an input feature set, to produce a different feature set of lower dimensionality in a way that maximizes the proportion of the total variance that is accounted for.

Because the variables are expressed in different units, centering and normalizing variables prior to clustering is recommendable. In other words, one performs the following transformation for x_{ij}, the i^{th} sample of the j^{th} variable:

$$x_{ij} \leftarrow \frac{x_{ij} - \mu_j}{\sigma_j} \tag{22}$$

where: μ_j and σ_j are respectively the average and standard deviation of variable X_j. Thus, all variables have identical variability and therefore the same influence in the computation of distances between samples.

In order to perform the mathematical technique called eigen analysis, we solve for the eigenvalues and eigenvectors of the square symmetric autocorrelation or scatter $d \times d$ matrix $S = [\sigma_{kl}]$, where σ_{kl} is the correlation between variables X_k and X_l and is defined as:

$$\sigma_{kl} = \sum_{i=1}^{N} x_{ik} x_{il} \tag{23}$$

The eigenvector associated with the largest eigenvalue has the same direction as the first principal component, and so on for each decreasing eigenvalue. Because the eigenvectors are orthogonal, each of theses eigenvalues is the inertia due to the corresponding principal factor, it is an index of the dispersion in the direction defined by the axis.

When one uses PCA in two dimensions, one can have a visual idea as to which groups appear naturally using only two dimensions to represent the feature data. Loosing information is sometimes necessary for a better understanding. The relative weight of the sum of the first two eigenvalues also describes how much information of the data structure PCA projection keeps.

Correlation between the PCA dimensions and previous dimensions can also help in detecting which are most significant. Pearson's linear correlation coefficient $R(X_k, X_l)$ between two random variables X and Y is defined as:

$$R(X_k, X_l) = \frac{\sum_i x_{ik} x_{il}}{\sum_i x_{ik}^2 \sum_i x_{il}^2} \tag{24}$$

3 Method and implementation

3.1 Creating benchmarks

3.1.1 Data sets

A variety synthetic data sets with compact and well-separated clusters of various shapes, sizes and dimensions were prepared. A common hurdle in most clustering algorithms is to identify and handle outliers (unusual data values, mostly due to data noise, rare events...). Outliers may cause unjust increase in the cluster size or a fault clustering altogether. Variants of data set 2 containing outliers were also used.

A real world data set, the `IRIS` data (see figure 6), was also used to test the clustering indexes performances. `IRIS` is a data set with 150 random samples of flowers from the iris species setosa, versicolor, and virginica collected by Anderson in 1935. From each species, there are 50 observations for sepal length, sepal width, petal length, and petal width in centimeters. One of the classes is clearly separable, whereas the other two overlap a little bit.

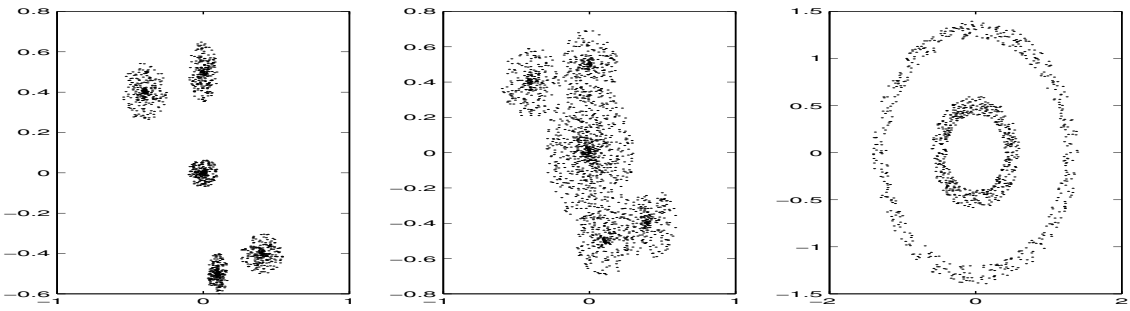

Figure 5: Synthetic data sets: 5 well-separated clusters, 5 overlapping, 2 rings

3.1.2 Tested clustering methods

The methods tested on the benchmark data sets were the K-means variants and GMMs, as well as DB, I, I_{fuz} (see [11, 15, 9]) and the partition coefficient defined as:

$$PC = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} u_{ik}^2 \tag{25}$$

indices were computed in `Matlab`. The closer the PC index is to unity, the crisper the clustering is. A `SVC toolbox` developed at Illinois University by Kliper, Pasternak and Borenstein was also used [1].

3.1.3 Results

Both DB and I indexes were able to find the correct number of clusters for the first type of data set (see table 3). Results were variable when clusters overlapped more, but in the case of data set 2, the correct number of groups is visually hard to infer. The recommended

[1] `http://www.cs.tau.ac.il/~borens/courses/ml/main.html`

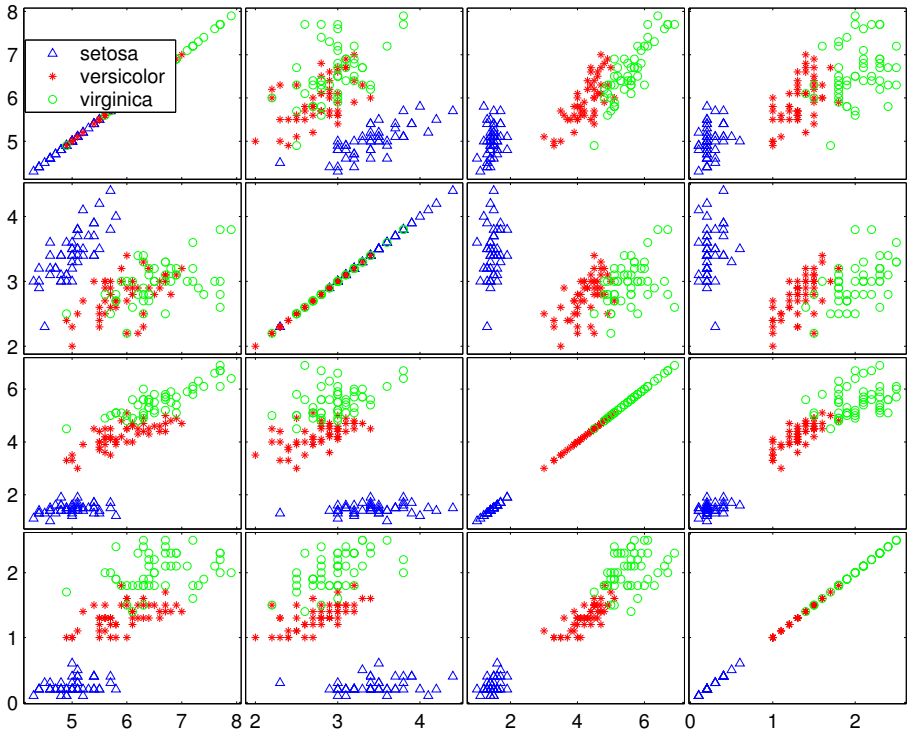

Figure 6: Fisher's Iris data set

Index	Data Set 1	Data Set 2	IRIS
DB	5	2	3
I	5	3	3
I_{fuz}	5	5	3
PC	5	3	3

Table 3: Sample data sets: recommended number of clusters

values for K in this case ranged from 2 to 5, which could have also been a possible answer for a human eye.

Though the partition coefficient PC had higher values when the data sets were clearly separated and results for the previous examples were good, its main drawback was its apparent monotonous dependency on the number of clusters K and on the fuzzy parameter b. This index was not kept as an indicator of the number of clusters, but only as a measure of cluster fuzziness.

K-means coupled with the previous two indexes yielded satisfactory results. Fuzzy-clustering coupled with the fuzzy index dealt better with overlapping clusters than its classical counterpart, as it can be seen by comparing results for data sets 1 and 2. An example of fuzzy clustering with results and membership values can be found in figure 7.

Using the `SVC` method was more delicate, since it required user given values for kernel parameters. In the case of `IRIS` data, the first class was very easy to separate from the

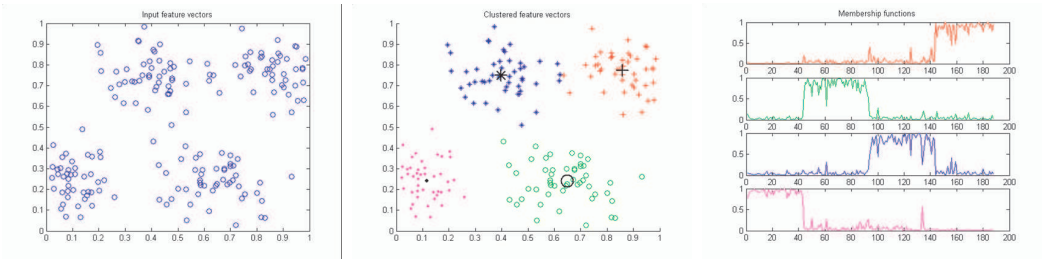

Figure 7: Synthetic fuzzy data set and clustering results

other two. The values of the parameters which gave best results for the other two classes were $q = 3$ and $C = 0.6$. The best values for separating the entire set were $q = 4.1$ and $C = 0.02$. In addition, `SVC` was, as expected, the only method able to cope with the ring-shaped clusters, whereas K-means did poorly. However, computational time was significantly increased with higher dimensionality and the number of support vectors became quickly dissuasive. Because of the high dimensionality of the speech data at stake and the probable strong overlapping of the different classes of phonemes, `SVC` was not used in the speech tests despite its ability to detect clusters with chain-like shapes.

Although clustering algorithms do not usually make use of the labels (or classifications) of the samples, they can still be used as classification algorithms. Assuming each cluster will include points with the same classification, we can determine the classification of each cluster by finding the classification majority within this cluster. With this method, if one partitions the data into two sets, one for clustering and one for testing, one can also predict the classification of a new sample that falls in the boundaries of each cluster and evaluate how good the clustering is.

3.2 Working with TIMIT speech data

3.2.1 The TIMIT speech corpus

The data considered was extracted from the `TIMIT` database, a corpus of read speech[2] (`*.wav`) with phoneme time-transcription of all sentences (`*.phn`). The speech database was designed to provide acoustic phonetic speech data for the development and evaluation of automatic speech recognition systems. `TIMIT` contains a total of 6300 sentences, 10 sentences spoken by each of 630 speakers from 8 major dialect regions of the United States.

3.2.2 Extracted features

A total of 27 coefficients were extracted using the `readHTKfeat.m` routine (A short description for each is provided in section 2). The first 24 were plain MFCC coefficients. Though the number of coefficients that are commonly used is 13, as well as first and second order time derivates, or Δ-*MFCC*, it was agreed from the start that all 24 MFCCs should be taken equally into account and that no Δ-MFCCs were to be used, in order not

[2]Intra-speaker variabilities due to emotion for example, were reduced because the recordings came from read text.

to add additional time correlation. It is often the dynamic characteristics of the features that provide most information about phonetic properties of speech sounds (related to, for example, formant transitions or the closures and releases of stop consonants). It will not be the case in this report, only plain MFCCs will be used. The initial aim was to see if groups of feature vectors obtained with clustering were consistent in time, if phases of a phoneme utterance could be spotted, etc. Energy, MACF ad ZCR were also used.

Performing K-means clustering correctly required having classes of approximately the same size. For this reason the experiments were performed on phonemes which were most often encountered in the data. Many phonemes had a very small number of occurrences. For example, consonants `g, z, b` and `t`, which are short by nature compared to certain vowels like `iy` or `ao` (see table 4). In each experiment, a small number of phonemes with at least 100 data samples was chosen (e.g. {`d,t,s,z,iy,eh,ao`}). The selection of phonemes was motivated by illustrating some of the acoustic properties, such as voicing, or articulatory properties, such as degree of openness. There seemed to be too little data for one speaker alone, so experiments were conducted over a given sentence read by several speakers of a given sex and dialect region.

3.2.3 Implementation

Routines allowed the user to specify:

- a list of sentences: `fcjf0/sa1` for sentence `sa1` of female speaker `fcjf0`
- a list of phonemes to extract
- how many feature dimensions should be kept
- how class sizes should be equalized, which features to reject
- parameters for algorithms
- if normalization should be done
- if PCA should be applied...

The clusterings, indexes and validity measures were computed for a number of clusters ranging from 2 to 15, performing 50 runs for each value of k. All cluster assignments were saved (`*.lab`) for display in time using `Wavefurfer`. The feature data extraction and clustering procedure are summarized in figure 8.

3.2.4 How many clusters?

It is obvious that a problem one faces in clustering is to decide the optimal number of clusters that fits a data set. When the classified data set and the clustering results were visualized in 2D-PCA projection, the degree of overlapping of the classes of phonemes was sometimes strong. Certain phonemes did not seem to cluster naturally into one group: stops had high intra-class dispersion, and overlapped classes containing fricatives, who were on the contrary more compact in nature.

It seemed unlikely that the estimated number of clusters would indicate the number of phonemes, but since certain acoustic and articulatory properties of speech were often

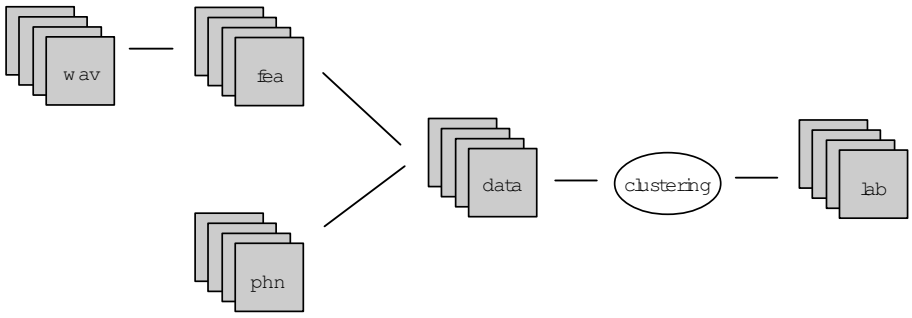

Figure 8: Feature data extraction and clustering procedure

common to points of the same cluster, perhaps indexes could reveal which of these properties were noticeable with this feature representation. The F-measure and purity (see [13, 14]) were calculated over all files.

4 Results

4.1 Example 1: broad phonetic classes {d,t,s,z,iy,eh,ao}

The first example contained two stops and two fricatives (voiced and unvoiced) as well as three vowels. The aim was to see how broad phonetic classes clustered together and how well the indexes would perform. A total of 100 feature vectors per phoneme, uttered by five different speakers, were extracted. The TIMIT transcription of these phonemes can be found in table 4.

Phonetic groups	Symbol	Example word	Possible phonetic transcription
stops	d	day	dcl d ey
	t	tea	tcl t iy
fricatives	s	see	s iy
	z	zone	z ow n
	v	van	v ae n
	f	fin	f ih n
vowels	iy	day	dcl d ey
	eh	tea	tcl t iy
	ao	bought	bcl b ao tcl t

Table 4: {d,t,s,z,iy,eh,ao}: TIMIT lexicon of phonemic and phonetic symbols

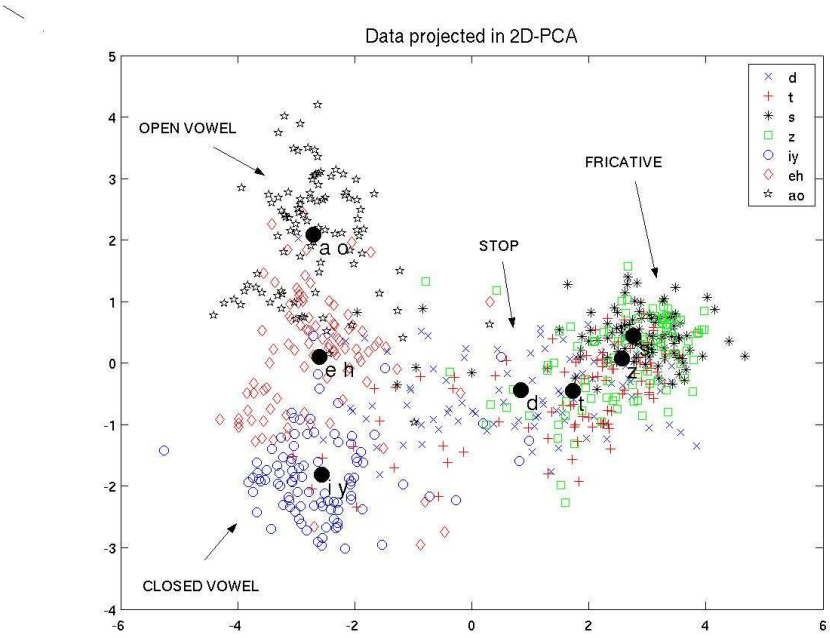

Figure 9: {d,t,s,z,iy,eh,ao}: data in 2D-PCA projection

PCA was applied to the data. The first two factors accounted for 46% of the total

scatter: normalized eigenvalues were $\lambda_1 = 0.35$ and $\lambda_2 = 0.11$ (see figure 10). Considering the number of variables, in our case 27, it is a fair amount. Figure 10 indicates there the data can be described with 5 to 7 linear factors (when eigenvalues λ become smaller than the average $\frac{1}{p}\sum_i \lambda_i$). Correlation between principal components and previous feature dimensions indicated that (energy + ZCR + MACF + first MFCC) and (third MFCC) were the most explicative factors for separating respectively consonants from vowels and different degrees of openness in vowels.

Figure 9 displays the feature data along the first two principal factors. Each feature vector is represented with a symbol: for instance, `[iy]` is a circle on the plot (bottom left). It is noticeable that certain groups of phonemes can be clustered: vowels (left) or consonants (right). Different degrees of openness opposing open vowels (top left) to closed vowels (bottom left) tend to produce separate clusters, with some overlapping for mid-open vowels. Phonemes `[s]` and `[z]` (right) are very disperse and have neighboring centroids, which makes them hard to separate. This is due to the fact `[z]` is often unvoiced in American English.

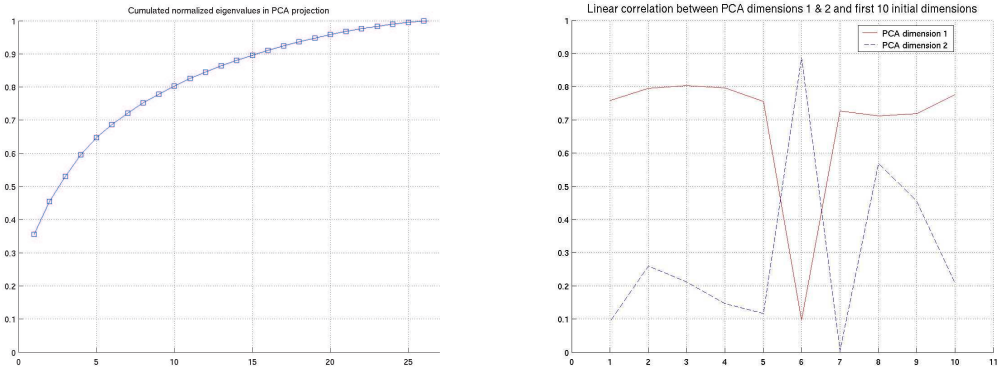

Figure 10: {d,t,s,z,iy,eh,ao}: cumulated scatter on principal factors (left), correlation between factors and feature dimensions (right)

On the whole, indexes and validity measures recommended variable numbers of clusters within a small range, mostly 2, 4 and 6. Figure 11 shows the clustering results on 2D-PCA data. The *DB* and *I* indexes recommended $k_{DB} = 2$ and $k_I = 4$, which resulted in clustering vowels and consonants on one hand, and degrees of openness, stops, fricatives on the other.

Clustering quality was compared for two versions of this data set. First all 27 dimensions were kept, then only the first two principal components. The F_{meas} ranged from 74% in the first case, to 62% with only two components but recommended 4 clusters in both cases. Purity for the entire clustering ranged from 69% to 59%. This would indicate that reducing the data to two variables still makes it possible to spot 4 different clusters which can be characterized in terms of acoustic and articulatory features.

In order to estimate clustering quality and usefulness, additional tests on data with the same phonemes from different sentences were conducted. Results were: 73% correct classification if the speakers were the same as in the training procedure, and as low as 40% when speakers from a different dialect region were involved. Confusion essentially arose

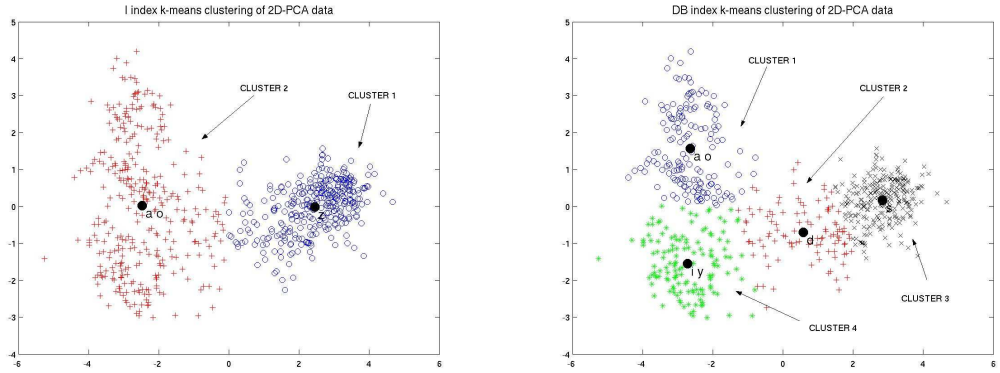

Figure 11: {d,t,s,z,iy,eh,ao}: I (left) and DB (right) index crisp clustering

between stops and fricatives, but vowels were always distinguished from consonants.

Broad phonetic classes, namely {consonants, vowels} or more precisely {stops, fricatives, open and closed vowels} were fairly well separated. Voicing was a strong discriminative factor : vowels and unvoiced consonants were in different regions, which sometimes did not overlap too much, though voiced consonants could be in either two depending on how strong the voicing was. Certain articulatory features, such as openness of vowels could also be differentiated. In addition, front and back vowels determined regions in the data, though middle vowels in between overlapped the previous two.

4.2 Example 2: vowels/consonants {s,f,v,iy}

The aim here was to test how the variables ZCR and MACF could discriminate phonemes. It appears from figure 12 that voicing was fairly well detected. It resulted in grouping voiced fricative [v] with the vowel [iy].

The number of clusters for both validity indices and measures (see figure 13) all agreed on the presence of 2 clusters. The speech data at stake here came from New England (dr1) female speakers. Results were identical using the same phonemes from different speaking styles, namely Northern (dr2) and Western (dr7). It appeared in all cases that these two dimensions were sufficient to separate voiced from unvoiced phonemes: {[v],[iy]} and {[s],[f]}.

The short-time autocorrelation function demonstrated its ability to reveal periodicity in a signal. On the other hand, the autocorrelation of the unvoiced speech segments [s] and [f] looks more like noise: phoneme [s] shows greater dispersion. In general, autocorrelation is considered as a robust indicator of periodicity.

Average zero crossing rate is a measure that allows discrimination between voiced and unvoiced regions of speech, or between speech and silence. Unvoiced speech has in general, higher zero-crossing rate.

When the cluster centroids were used as a K-means classifiers, additional voiced and unvoiced phonemes were distinguished fairly well: in 80% of the cases for voiced phonemes and 62% for unvoiced phonemes. The voiced phoneme [z] was the one which caused the most confusion. Purity of the two resulting clusters ranged from 85% for voiced phonemes to 80% in the case of unvoiced phonemes.

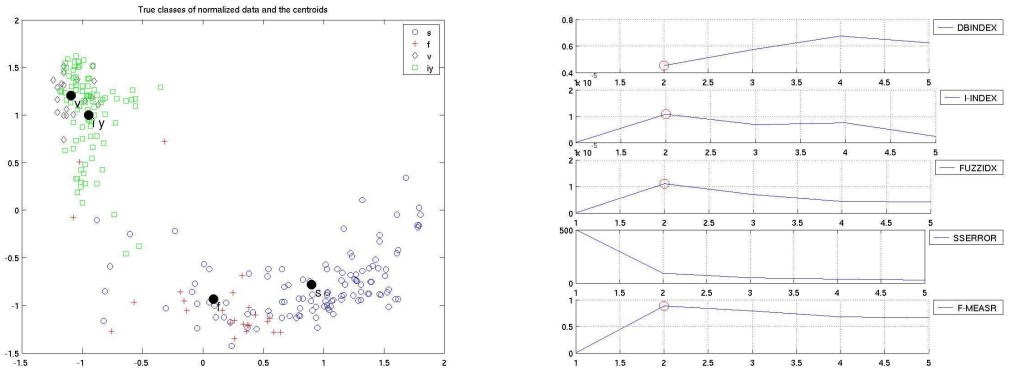

Figure 12: {s,f,v,iy}: data in 2D-PCA projection (left) and indices (right)

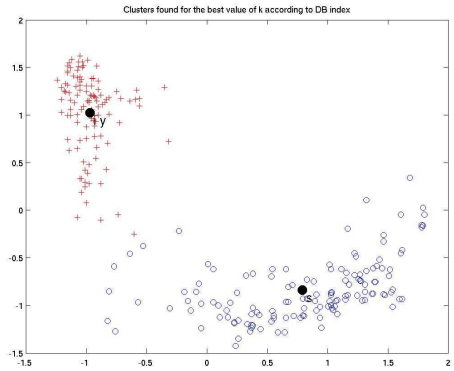

Figure 13: {s,f,v,iy}: *DB* clustering results

4.3 Example 3: front/open vowels {ae,ah,aa}

Among the various acoustic and articulatory features used to classify phonemes, the degree of openness for vowels seemed an interesting one to investigate upon. The previous example illustrated that the third MFCC coefficient did well in distinguishing open from closed vowels, so the idea here was to try to improve separation of these vowels using the previous methods as well as fuzzy clustering, because of the high degree of overlapping.

Phonetic groups	Symbol	Example word	Possible phonetic transcription
front vowel	ae	day	dcl d ey
	ah	but	bcl b ah tcl t
back vowel	aa	bott	bcl b aa tcl t

Table 5: {ae,ah,aa}: TIMIT lexicon of phonemic and phonetic symbols

Here we consider three phones, front to back: the front vowel [ae], [ah], and the back vowel [aa].

In this case, there seemed to be no clear clusters in 2D-PCA (see figure 14), so all feature dimensions were kept.

Phoneme	F_1	F_2	F_3
ae	660	1720	2410
ah	520	1190	2390
aa	730	1090	2240

Table 6: {ae,ah,aa}: formant frequencies of vowels

These three vowels differ in their first three formant frequencies, especially for F_2, which is higher for front vowels such as [ae]. From figure 16, we see that discrimination between different degrees of openness can be explained with MFCC coefficients 1 and 3.

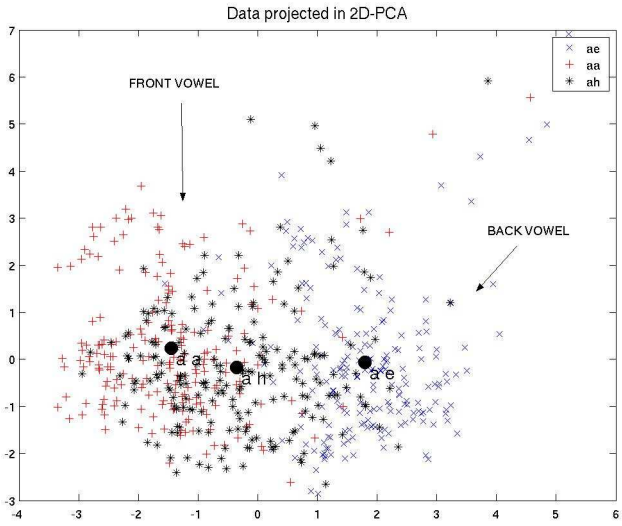

Figure 14: {ae,ah,aa}: feature data in 2D-PCA

The recommended number of clusters by the I_{fuz} index was 2. Figure 15 displays the clustering results. The histogram on the left indicates how each phoneme was clustered: Phonemes [ae] and [aa] were clustered separately with recall values of 95%. The histogram on the right hand side shows the content of both clusters: precision remains rather low since [ah] is present in both clusters. This resulted in a global purity of 70% for the entire clustering. When phoneme [ah] was not considered, separation between front and back vowels [ae] and [aa] was good: 95% correct clustering and the classifier was able to separate additional phones pronounced by different speakers of the same sex with an efficiency of over 90%.

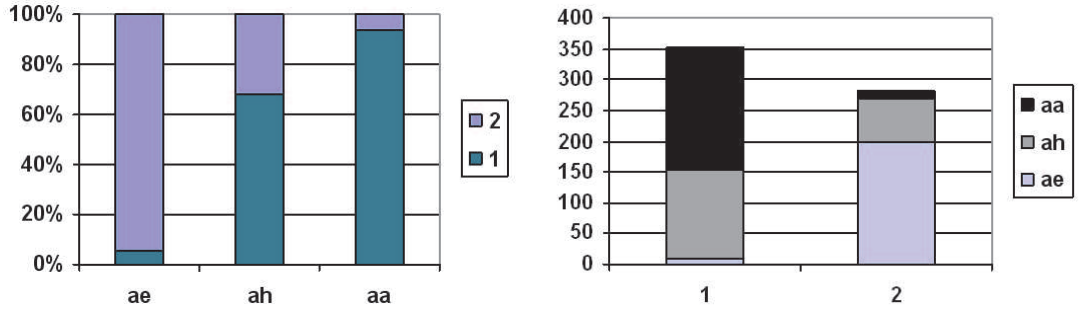

Figure 15: {ae,ah,aa}: content of the clusters

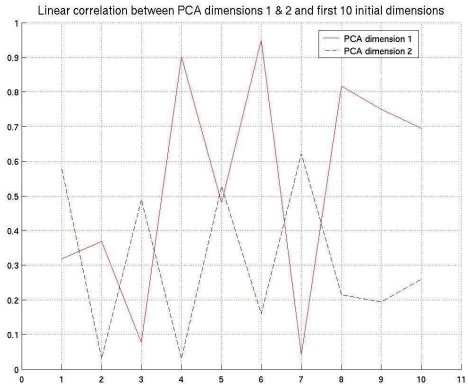

Figure 16: {ae,ah,aa}: principal comp. correlation with feature dimensions

4.4 Example 4: fricatives {s,z,v,f}

The aim here was to discriminate fricatives and see which features differentiate them in order of appearance. Formant transitions at vowel-consonant boundaries may be crucial in identifying the consonant. Since, for many consonants, such transitions are very rapid, they do not occupy many frames. This led to extracting consonants which were bordered by similar vowels (open to be more specific).

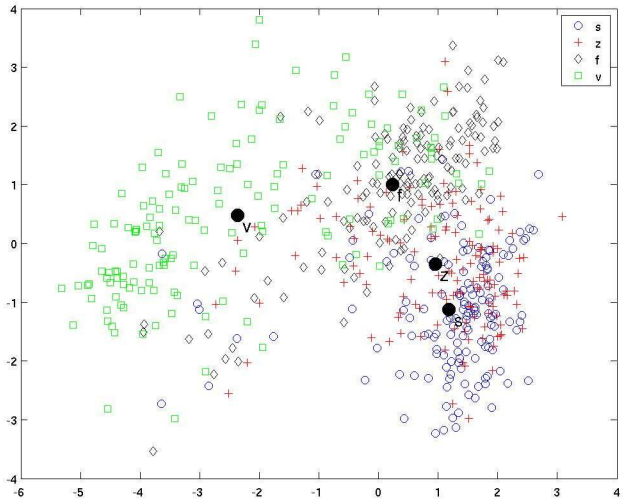

Figure 17: {s,z,v,f}: feature data in 2D-PCA

Linear correlations were weak in this case, correlation coefficient values remained below 80% (see figure 18). Distinction along factor 1 between [v] and the other three phonemes, in other words in terms of voicing, was due to (MACF+ZCR) as well as (MFCC coefficients 1 and 7). Factor 2 was mostly correlated to energy and MFCC coefficient 4. This factor

was able to distinguish between labio-dentals [v],[f] from [s],[z], which are not.

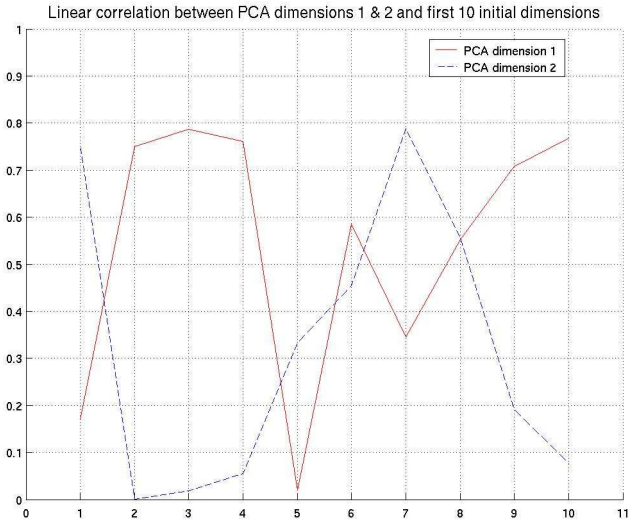

Figure 18: {s,z,v,f}: principal comp. correlation with feature dimensions

Indexes	Recommended K
DB	2
I	3
I_{fuz}	3
Measures	
F_{meas}	3
ρ	3

Table 7: {s,z,v,f}: Best K for each index and measure

Crisp and fuzzy indexes all recommended 2 or 3 clusters (see table 7). Constraint-based clustering, using fuzzy K-means with the I_{fuz} index and a threshold at 80% on the membership values reduced the co-articulation effect between feature vectors from different phonemes, at vowel-consonant transitions for example. This procedure also increased cluster quality in terms of precision, recall and purity, as depicted in histograms from figure 19. The top figures display clustering results according to the I index and the bottom figures correspond to results for I_{fuz} with the rejection criterion. The rejected feature points are grouped under rej in the histograms. Clustering results for phoneme [z] for instance were improved using this method. Clusters showed to be more precise with respect to the phoneme they best represented: cluster 3 captured more [f] in the fuzzy method that it did in it crisp counterpart.

When displayed in Wavesurfer, cluster assignments were stable in time. The beginning of the spectrum corresponding to sentence si2036 is displayed in figure 20. The words are "as his feet". In figure 21, each line indicates the assignments according to crisp

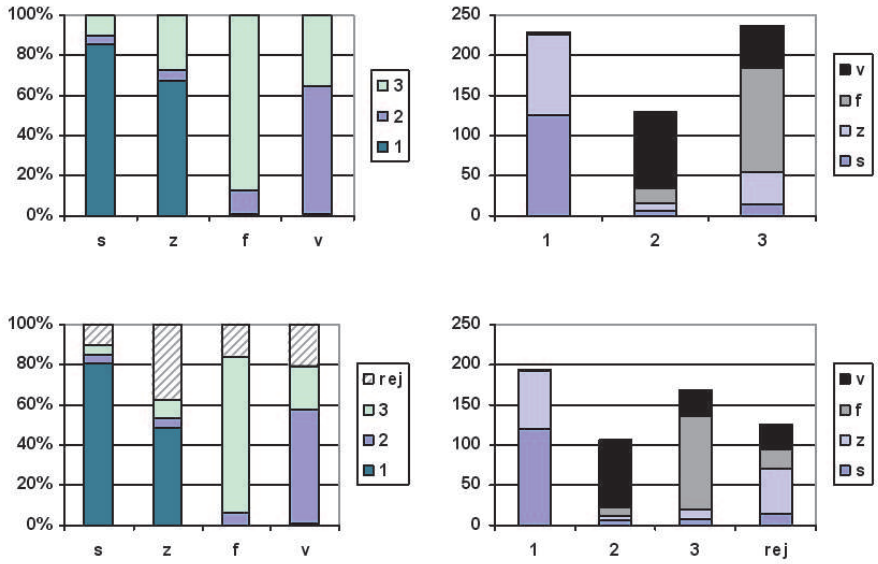

Figure 19: {s,z,v,f}: crisp (top) and fuzzy (bottom) clustering results

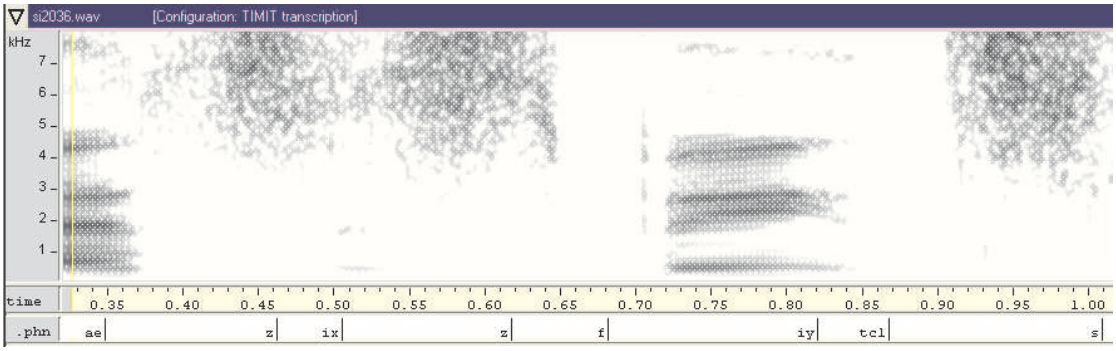

Figure 20: {s,z,v,f}: speech spectrum for si2036

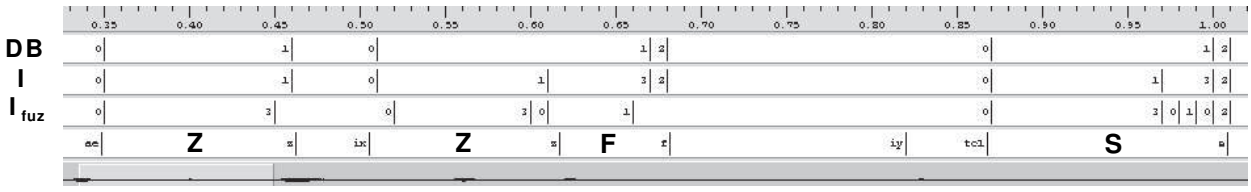

Figure 21: {s,z,v,f}: cluster assignments in time for si2036

partitioning using the *DB* and *I* indexes for the first two and according to I_{fuz} with a threshold on the membership values set at 80%. The numbers correspond to the cluster assignments and 0 stands for points which were not considered for the experiment, or rejected in the fuzzy case.

All three results indicate that phonemes [s] and [z] were not distinguished. Increasing the number of clusters did not yield better results.

DB recommends two clusters and assigns [z] and [f] in cluster 1 and transition feature vectors in 2. *I* on the other hand recommends three which makes it possible to differentiate the two (1 for [z] and 3 for [f], whereas 2 corresponds to transition feature vectors). The fuzzy method rejected many feature vectors in phone to phone transitions (clustered as 0). Consider phoneme [f] and its transition with phoneme [iy]: figure 21 shows the co-articulation effect which the fuzzy method was able to detect. It also resulted in eliminating points which laid on the borders in the transition [z] → [ix] → [z]. Figure 20 shows the continuous variations in the spectrum, which makes the borders between phonemes hard to define.

PHONEME	GRAVE	VOICED	CONTINUANT	STRIDENT
s	−	−	+	+
z	−	+	+	+
v	+	+	−	−
f	+	−	+	+

Table 8: {s,z,v,f}: Binary feature classification

Jackobson and Halle's binary feature classification for the phonemes is displayed in table 8. All other features features were identical. Clustering results indicate a possible ranking of the features. [v] was the easiest to separate because of [+voiced]. Then came [f], which, unlike [s] and [z], is [+grave]. Though [z] is also considered as voiced, the spectrum from figure 20 shows that this property is not very pronounced in English. From the binary feature point of view, there is no difference between [z] and [s], which would explain why it was impossible to distinguish the two when using clustering methods. However, other attempts to rank different features which did not comprise [+voiced] did not always yield the same rankings. Perhaps an average ranking can be performed over several sets, but the methods used here are not suitable.

5 Discussion

Visualization of the data using PCA indicated that phonetic clusters overlapped strongly, but that broad phonetic classes could be clustered well. In addition, when phonemes from a given broad phonetic class (e.g.: fricatives) were clustered, the most discriminative dimensions were energy, (zero-crossing rate + maximum of the autocorrelation function + a few MFCC coefficients), and clustering using only these dimensions sometimes yielded similar results. Interpretation of the MFCC coefficients remained difficult.

The algorithms used in the experiments were able to cluster some of the overlapping broad phonetic classes and distinctive binary features. In order to reduce dependency on initialization, GMMs using the EM algorithm were used prior to K-means and the clusterings were more stable than without this procedure over different runs.

Cluster assignments were relatively stable in time for a given speaker. Fuzzy K-means coupled with I_{fuz} were able to detect co-articulation effects on the feature vectors. It is likely though, that using Δ-MFCC would have made it possible in a simpler way.

Indexes did not yield the number of phonemes. However, the I_{fuz} index often recommended the same number of clusters as the F_{meas}. In most experiments, broad phonetic classes (e.g.: stops, fricatives, open/closed vowels, etc.) were fairly captured by the clustering methods used.

F_{meas} remained below 70% in most cases. This measure describes how well K-means clustering can fit the classification and, considering the number of outliers, 70% is perhaps a good result. In addition, F_{meas} and purity were often maximized for the same number of clusters. In other words, the optimal value for the number of clusters k is such that the clustering has the best average precision and recall over all other clusterings, and its clusters redefine the classification as well as possible with this method.

On the whole, the methods used in this report were relatively simple but were able to illustrate articulatory and acoustic differences of phonemes, and did fairly well in computing classifiers. It is difficult however to conclude that this procedure conducted for the various experiments would work well in general.

More data for one speaker would be recommendable. Though the recorded `TIMIT` speech was read and that speakers had identical sex and dialect region, speaker variability was also accountable for the overlapping of the clusters.

Perhaps the phonetic clusters do not exactly have the sought convex shapes, say chain-like shapes for instance. Support Vector Clustering could better deal with outliers and overlapping clusters, without assuming the sizes and shapes of the clusters it is supposed to find. Also, the metric used here was the plain Euclidean distance. A different metric should also we tested: for example one determined by the covariance matrices of the determined clusters.

6 Conclusion and further work

An attempt to rank all 14 binary features from Jackobson and Halle's classification should be performed. Hierarchical tree clustering could possibly create it. We have seen that principal component analysis illustrated that certain feature dimensions had a stronger discrimative power than others for phoneme distinction. Unfortunately, this method is more *descriptive* than it actually helps in *deciding* which phoneme a feature vector represents. It is likely that discriminative analysis would be more suitable in this case.

For example, one could compare phonemes that differ by a reduced number of binary features in order to rank them by discriminative power. This would produce a set of pairwise rankings. Then in order to perform a global ranking, we could seek an optimal ranking that represents as much the partial ones as possible. Discriminative analysis would also allow one to reduce the number of features from Jackobson's classification to those who create a significative difference between phonemes.

This procedure could give an insight as to which characteristics of phonemes children first learn to differentiate as they learn to speak. Moreover, it would be interesting to cluster baby speech into groups of patterns of phonemes and study the level of phonetic feature discrimination of the child's speech.

References

[1] D. Anguita, S. Ridella, F. Rivieccio, and R. Zunino. Unsupervised clustering and the capacity of support vector machines. In *Proc. of the IEEE Int. Joint Conf. on Neural Networks*, 2004.

[2] A. Ben-Hur, D. Horn, H. Siegelmann, and V. Vapnik. Support vector clustering. *Journal of Machine Learning Research*, 2:125–137, 2001.

[3] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.

[4] V. Estivill-Castro. Why so many clustering algorithms: a position paper. *SIGKDD Explorations*, 4(1):65–75, 2002.

[5] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Clustering algorithms and validity measures. *IEEE Transactions on pattern analysis and machine intelligence*, 24(12), 2002.

[6] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Clustering validity checking methods: part ii. *SIGMOD Rec.*, 31(3):19–27, 2002.

[7] L. Hyman. *Phonology: Theory and Analysis*, chapter 3. New York: Holt, Rinehart and Winston, 1975.

[8] J. Jantzen. Neurofuzzy modelling. Technical Report 98-H-874, Technical University of Denmark: Oersted-DTU, 1998.

[9] U. Maulik and S. Bandyopadhyay. Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12), 2002.

[10] M. Meila and D. Hecherman. An experimental comparison of model-based clustering methods. *Microsoft Research, Machine Learning*, 42:9–29, 2001.

[11] G. Milligan and M. Cooper. An examination of indexes for determining the number of clusters in a data set. *Psychometrika*, 50:159–179, 1985.

[12] G. Richard. Traitement de la parole. *ENST, brique PAMU*, 2003.

[13] M. Rosell, V. Kann, and J.-E. Litton. Comparing comparisons : Document clustering evaluation using two manual classifications. In *ICON*, 2004.

[14] B. Stein, S. Meyer, and F. Wissbrock. On cluster validity and the information need of users. In *Third International Conference on Artificial Intelligence and Applications*, pages 216–221, 2003.

[15] M.-C. Su. A new index of cluster validity. presentation, National Central University of Taiwan, 2004.

[16] A. Ultsch and C. Vetter. Self-organizing feature maps versus statistical clustering methods. Technical Report 0994, Universiy of Marburg, 2001.