# IS SPEECH DATA CLUSTERED? - STATISTICAL ANALYSIS OF CEPSTRAL FEATURES

*Tomi Kinnunen, Ismo Kärkkäinen and Pasi Fränti*
{tkinnu,iak,franti}@cs.joensuu.fi

*University of Joensuu, Department of Computer Science,*
*P.O. Box 111, 80101 JOENSUU, FINLAND*

***Abstract:*** Speech analysis applications are typically based on short-term spectral analysis of the speech signal. Feature extraction process outputs one feature vector per frame. The features are further processed by application-dependent techniques, such as hidden Markov models or vector quantization. Independent from the application, it is often desirable that the feature vectors form separable *clusters* in the feature space. In this work, we study whether data is really clustered in the feature space and, if so, what is the number of the clusters in typical speech data. We consider different forms of the widely used cepstral features.

*Keywords:* Speech analysis, pattern recognition, short-term features, cluster analysis.

## 1 INTRODUCTION

In speech analysis, parametric description of the signal is needed to characterize its acoustical properties. *Feature extraction* is a process that computes certain *feature vectors* from the speech signal by short-term spectral analysis techniques. For each short frame of speech (e.g., 20 ms) the extraction procedure outputs a feature vector that describes the acoustical characteristics of this frame.

The selection of the features varies from application to another. However, it is generally desired that dissimilar acoustic vectors would be clearly separable from each other in the feature space and correspondingly, similar vectors would be close to each other. In terms of pattern recognition, the inter-class variances of the vectors should be large and intra-class variances should be small for good recognition accuracy.

Due to the above reasons, it is desirable that the feature vectors would form separable *clusters* in the feature space. However, detailed analysis of the feature vectors does not support this assumption. For example, we can examine the data by transforming the multi-dimensional feature vectors to two-dimensional parameter space using principal component analysis (PCA) and perform visual examination of the data. In our experiments, we found out no evidence of separable clusters in our data sets. Dimensional reduction of the data, on the other hand, can loose information of the data and, therefore, this kind of analysis cannot confirm the existence or absence of separable clusters.

In addition to the above examination, we have also evaluated the performance of different clustering algorithms for the speaker identification problem in [5]. We used the lowest 12 mel-cepstral coefficients as the features. We found out that different clustering algorithms gave virtually similar recognition accuracy, and that the accuracy was monotonically increasing with the size of the codebook used in vector quantization (VQ).

These above observations indicate that there are no separable clusters in the data. In this work, our goal is to verify this hypothesis. We proceed by performing statistical analysis of the distributions of the common feature vectors used in speech processing. We apply cluster analysis methods for finding out how many clusters there are in the data set. If the analysis shows that there are several clusters in the data, the clustering structure and the knowledge about the number of clusters could be exploited in the recognition. On the other hand, if there are no clusters, we can conclude that the role of vector quantization is merely to reduce the amount of the feature data and being a tool for the recognition process.

The rest of the paper is organized as follows. In Section 2, we give a short description of the features used in this study. In Section 3, we describe the specialized distance metrics and a normalization method applied for the feature vectors. In Section 4, we describe the clustering process and the criterion used in the determination of the number of clusters. Test setup and results from the experiments are given in Section 5. Conclusions from the results are drawn in Section 6.

## 2 THE FEATURES

Feature extraction of speech signal is usually based on short-term spectral analysis. This means that the signal is divided in short, fixed-length *frames*. The adjacent frames are usually overlapping (e.g., by 50% of the length of the frame) and each frame is multiplied with a smooth *window function* to avoid ”spectral artifacts” caused by discontinuities in the endpoints of the frame. The most popular window function in speech processing is the *Hamming* window, see, e.g. [3].

We consider the three alternative features:

- *Real cepstral coefficients*
- *Mel-frequency cepstral coefficients*
- *Linear predictive cepstral coefficients*

***Real cepstral coefficients (RCC):*** *Cepstrum* [2] is a parametric representation for the envelope structure of the short-term speech spectrum [3]. The envelope of the spectrum is mainly due to the resonant frequencies of the vocal tract called *formants*. Therefore, a cepstal vector characterizes the shape of the vocal tract at the current frame. Real cepstrum is computed as inverse Fourier transform of the logarithm of the magnitude spectrum [3]:

$$RCC(n) = FFT^{-1}(\log|FFT(s(n))|), \quad (1)$$

where $s(n)$ denotes the frame over which the cepstrum is computed.

***Mel-frequency cepstral coefficients (MFCC):*** *Mel-cepstrum* is computed closely in the same way as RCC. In addition to RCC, it uses psychoacoustical weighting in the frequency domain before the inverse FFT. Details of the MFCC computation can be found in general speech processing books, such as [3].

***Linear predictive cepstral coefficients (LPCC):*** Another way to compute the cepstral coefficients is to do it via *linear predictive analysis (LPA)*; for the details of LPA, see, e.g. [3]. Given the linear predictive coefficients $a_k$, $k$=1,...,$N$, the LPCC are determined by the following recursive relationship [1]:

$$\begin{cases} c_1 = a_1 \\ c_n = \sum_{k=1}^{n-1}(1-\frac{k}{n})a_k c_{n-k} + a_n, \quad n = 1,...,P \end{cases} \quad (2)$$

where $P \le N$ is the desired number of cepstral coefficients.

## 3 FEATURE DOMAIN NORMALIZATION

An important issue in any pattern analysis problem is the correct choice of the *distance measure* in the feature space. The selection of the measure depends on the vectors and therefore, requires knowledge of their nature.

Let us denote $\boldsymbol{x} = (x_1, x_2, ..., x_P)^T \in \mathbf{R}^P$ as a feature vector and $P$ as the dimension of the feature vector space. By *distance measure*, or *metric*, we refer to a function $d : \mathbf{R}^P \times \mathbf{R}^P \to \mathbf{R}$ which measures the dissimilarity between any two feature vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. For identical vectors $d(\boldsymbol{x}, \boldsymbol{y}) = 0$.

***Euclidean distance:*** To our knowledge, there is no clear consensus in the literature about what is the correct way to measure the distance of cepstral vectors (RCC, MFCC or LPCC). The Euclidean distance, on the other hand, corresponds best to our intuitive conception of physical distance between two points and with a proper normalization, it is still quite useful. It is defined as:

$$d_E(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x}-\boldsymbol{y})^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{y})} = \sqrt{\sum_{k=1}^{P}(x_i - y_i)^2} \quad (3)$$

***Normalization:*** The basic problem with the Euclidean distance is that it is *scaling variant*. This means that if the variances of the vector components differ much from each other, the components with large variance dominate the distance value. For example, the variances of the vector components in a typical MFCC vector vary greatly. Normalization of these vectors is therefore needed. We consider the following simple normalization technique. A special case of the widely used *Mahalanobis-distance* [3] can be obtained using the following normalization:

$$x_k' = \frac{x_k - \mu_k}{\sigma_k} \quad \forall\, k = 1,..., P. \quad (4)$$

Here $x_k$ and $x_k'$ are the original and normalized vector components, respectively, and $\mu_k$ and $\sigma_k$ are the mean and standard deviations of the $k$th component over all input vectors. The vectors have zero mean and unit variance after the normalization.

## 4 CLUSTER ANALYSIS OF THE FEATURES

The goal of the cluster analysis is to determine how many clusters there are in the data, and to identify these clusters. We denote the number of clusters by $M$, and represent the clusters as the set of their centroids $\{\boldsymbol{c}_i\}$. We use the *randomized local search* (*RLS*) algorithm [4] for solving the location of the clusters as it provides better and more reliable clustering results than the standard GLA or LBG algorithm [7]. The RLS algorithm is outlined in Figure 1.

In searching the clusters and their correct number, we use the following technique: For each number of clusters in a given range [1, $M$], we seek a good solution using the randomized local search and then select the best solution according to some predefined *criterion*. The criterion we chose to use is based on *F-test* [6].

The F-test measures statistical significance of the hypothesis that the variances of two given Gaussian distributions are different. Here we assume that the feature vectors $\boldsymbol{x}_i$ are the samples and that they are scattered around the clusters with Gaussian distribution.

Given the clustering as a set of cluster centroids $\{\boldsymbol{c}_i\}$ we can estimate the variance by measuring the sum of the square distances (*SE*) of the data vectors $\{\boldsymbol{x}_i\}$ and the centroids:

$$SE = \sum_{i=1}^{N} d(\boldsymbol{x}_i, \boldsymbol{c}_{g_i})^2, \quad (5)$$

where $d$ is the Euclidean distance, and the indices $g_i$ represent the partition of the feature vectors into the clusters. The comparative value is the variance of the entire data set representing the situation where there are no clusters. The variance can be calculated as:

$$\bar{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i \qquad Q_B = \sum_{i=1}^{M} n_i d(\boldsymbol{c}_i, \bar{\boldsymbol{x}})^2 \quad (6)$$

where $\bar{x}$ is the centroid of the data set. The F-test can now be calculated as:

$$F = \frac{SE \cdot (M-1)}{Q_B \cdot (N-M)}. \quad (7)$$

The smaller the value, the higher indication it is that the two variances are from different sources.

We then measure the F-test value for all clusterings we have generated. The clustering with the smallest value corresponds to the clustering with the correct number of clusters.

```
RLS(X, C, G) return C, G
FOR all i ∈ [1, N] DO g_i ← j such that x_i is nearest to c_j;
FOR a ← 1 TO NumberOfIterations DO
    C_new ← C;
    c_new,j ← Randomly chosen data vector;
    C_new, P_new ← GLA(X, C_new);
    IF F(X, C_new, G_new) < F(X, C, G) THEN
       C ← C_new; G ← G_new;
    END IF;
END FOR;
Return C, G;
```

*Figure 1: Pseudocode for the randomized local search.*

## 5 EXPERIMENTAL RESULTS

We collected 7:55 minutes of spontaneuous speech from one male speaker in a laboratory environment with a PC computer. All recordings were done in the same session. The speaker was prompted to read the same 12 sentences in a steady, clear voice, six times each. We designed the contents of the sentences so that they contain as many different phonemes as possible. In Finnish language, there are approximately 45 phonemes.

The speech files were sampled at 8 kHz with 16-bit resolution. Before the feature extraction, the following pre-processing steps were performed:

- Remove the DC offset (average substracting)
- Remove silent parts using simple short-term energy based thresholding.
- High emphasis with a filter $H(z) = 1 - 0.97z^{-1}$.

After silence removal, the length of the speech data was 5:13 minutes.

In the feature analysis, we varied the parameters as shown in Table 1. The extracted features are plotted in Figures 2-3 by reducing the 10- or 12-D vectors into 2D-space using *principal component analysis (PCA)*. In these examples, about 80-90 % of the energy of the original vectors is preserved in the first two eigenvectors. The illustrations indicates that if there are clustering structures in the data set, it cannot be extracted to the first two eigenvalues of the PCA.

*Table 1: Parameters of the feature extraction.*

**Features:**

- 12 lowest RCC coefficients,
- 12 lowest MFCC coefficients ($c_0$ dropped away),
- 10 LPCC coefficients from 10$^{th}$ order LP analysis.

**Windowing:** *Hamming window,*

- Size = 30 ms, shifted by 15 ms (50% overlap),
- Size = 15 ms, shifted by 5 ms (33% overlap).

**Normalization:**

- No normalization
- Normalization using Eq. (4)

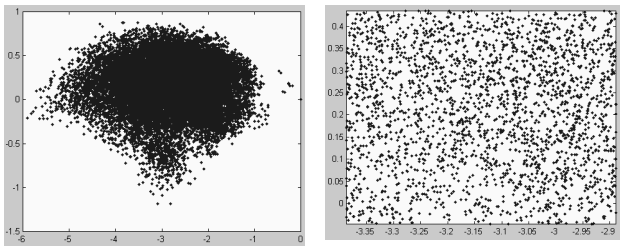

*Fig. 2: Visualization of the RCC data (right: zoomed).*

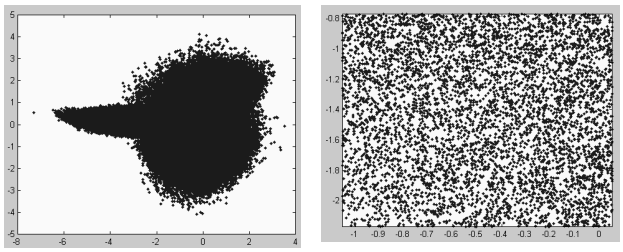

*Fig.3: Visualization of the LPCC data (right: zoomed).*

The results of the statistical cluster analysis are shown in Figures 4-6 for the normalized RCC, MFCC and LPCC features. From the F-test curves we see the ascending trend without global minimum in the cluster range [2, *M*]. The results indicate that there are no separable clusters in the feature space. The observation is similar for all tested features, and independent of the windowing parameters. Furthermore, the use of normalization did not make any difference and therefore, only the results for the normalized features are presented here.

To sum up, both the visual illustration of the two principal components, and the statistical cluster analysis indicate that there are no clusters in the data. The distribution of the feature vectors should therefore be considered more or less as a continuous probability distribution, than a set of data clusters.

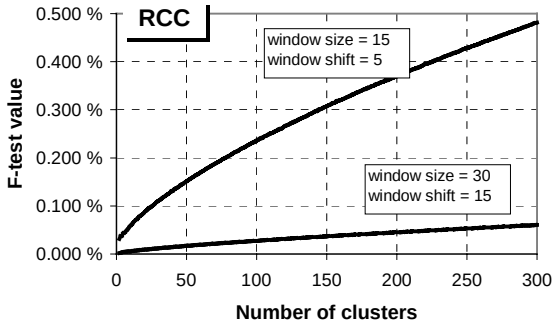


*Fig.4:Clustering results for the RCC parameters.*

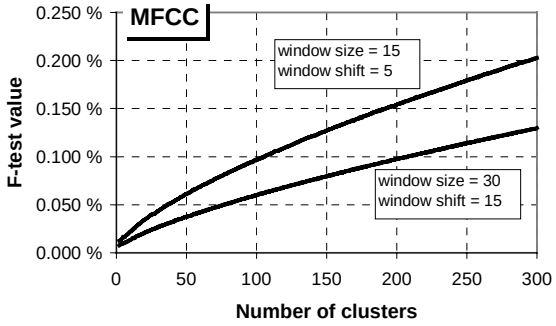


*Fig.5: Clustering results for the MFCC parameters.*

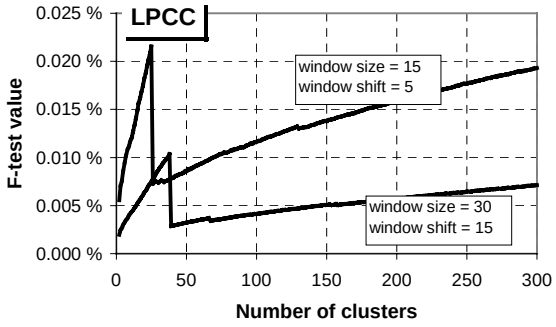


*Fig.6: Clustering results for the LPCC parameters.*

## 6 CONCLUSIONS

In this work, we have confirmed our hypothesis that using cepstrum-based parametrization of the speech signal, there are no clusters in the (Euclidean) feature space. Allthough this has already been accepted by several authors in the speech research field, there are still confusion with the terminology. The term “clustering” is still widely used even if it gives incorrect implication that the data were clustered, which is not the case according to our studies.

An important implication of our results is that when dealing with speech applications, the role of the vector quantization is to reduce the amount of data, and to model the distribution of the feature vectors. The ability of detecting clusters, on the other hand, is not an important property in the codebook generation. Thus, any fast algorithm that picks codevectors uniformly among the sample feature vectors can be used.

## REFERENCES

[1] Atal B.S.: ”Effectiveness of linear prediction charasteristics of the speech wave for automatic speaker identification and verification”, *Journal of the Acoustical Society of America*, **55**(6): 1304-1312, 1974.

[2] Bogert B.P., Healy M.J.R., Tukey J.W.: “The quefrency alanysis of time series for echoes: cepstrum, pseudo-autocovariance, cross-cepstrum, and saphe cracking”, *Proc. Symposium Time Series Analysis:* 209-243. New York, 1963.

[3] Deller Jr. J.R., Hansen J.H.L., Proakis J.G.: *Discrete-time Processing of Speech Signals*. Macmillan Publishing Company, New York, 2000.

[4] Fränti P., Kivijärvi J.: “Randomized local search algorithm for the clustering problem”, *Pattern Analysis and Applications*, **3**(4), 358-369, 2000.

[5] Kinnunen T., Kilpeläinen T., Fränti P.: ”Comparison of clustering algorithms in speaker identification”, *Proc. IASTED Int. Conf. Signal Processing and Communications (SPC)*: 222-227. Marbella, Spain, 2000.

[6] Krishnaiah P.R.: *Handbook of Statistics 1: Analysis of Variance*. North-Holland Publishing Company, Amsterdam, 1980.

[7] Linde Y., Buzo A., Gray R.M.: “An algorithm for vector quantizer design”. *IEEE Trans. on Communications*, **28**(1): 84-95, 1980.