ON HIERARCHICAL CLUSTERING FOR SPEECH PHONETIC SEGMENTATION

Ciro Gracia, Xavier Binefa

Universitat Pompeu Fabra, Department of Information and Communications Technologies, Spain

ABSTRACT

In this paper, we face the problem of phonetic segmentation under the hierarchical clustering framework. We extend the framework with an unsupervised segmentation algorithm based on a divisive clustering technique and compare both approaches: agglomerative nesting (Bottom-up) against divisive analysis (Top-down). As both approaches require prior knowledge of the number of segments to be estimated, we present a stopping criterion in order to make these algorithms become standalone. This criterion provides an estimation of the underlying number of segments inside the speech acoustic data. The evaluation of both approaches using the stopping criterion reveals good compromise between boundary estimation (Hit rate) and number of segments estimation (over-under segmentation).

Index Terms: speech segmentation, hierarchical clustering, stopping criterion.

1. INTRODUCTION

In this paper we address the problem of speech segmentation. Our fundamental objective is to define a procedure for speech segmentation in such a way that segments capture the fundamental spectro-temporal patterns inside the acoustic data. However, defining and validating this objective is not an easy task so instead we assumed that phonetic segmentation yields segmentation of underlying structures in the speech data.

Phonetic segmentation is the action of dividing the speech signal into its basic language functional units: the phonemes. The main issue is that manual phonetic transcription is an expensive task. Segmenting and labeling phonemes on large speech data is a laborious task that requires expert knowledge and incorporates the human factor. In addition phonetic segmentation is a processing step depended upon by other speech processing or recognition applications. For instance speech recognition systems require transcription of a large data corpus in order to build acoustic models and speech coding benefits from phonetic segmentation as it provides extra information for optimal coding. Previous approaches on literature regarding automatic phonetic segmentation can be divided into two categories: temporal domain and spectral domain. Approaches using signal time domain representation [4] preserve temporal information and do not introduce distortion due to spectral analysis, especially on the phoneme transition regions, where the stationarity property of the signal does not hold. However, many time domain algorithms for segmentation require windowing of the signal like most spectral analysis, but do not provide a 2D representation (spectro-temporal) of the signal which is especially meaningful in recognition of overlapped acoustic events [6]. In spectral domain we can differentiate between algorithms defining phonetic boundaries where signal characteristics exhibit an abrupt change [2, 10] and the algorithms which define segmentation as an optimization problem based on a certain goodness criteria [8].

We follow a similar approach to [8] which uses a clustering method in order to estimate the optimal speech segmentation given an objective function. This approach provides a framework that exhibits two advantages: in one hand, the objective function can be easily modified and, in second hand, prior knowledge can be easily included. Despite of these advantages, the algorithm lacks on being totally blind since it requires prior knowledge of the number of segments to be estimated. However, There are many applications requiring speech segmentation where there is no prior knowledge of the number of segments inside the data. In this paper we deal with this limitation by extending the algorithm in order to become totally blind.

2. UNSUPERVISED OPTIMAL PHONETIC SEGMENTATION

Previous work [8] faced the problem of defining the partition of a continuous speech signal into contiguous, non overlapping segments that maximizes an objective function. In order to estimated this optimal segmentation in an efficient way, the optimization process can be faced by a hierarchical agglomerative nesting algorithm. The algorithm takes into account the contiguous property of the segments, which means that only grouping operations that result into contiguous segments will be taken into account. Different objective criterion was presented but most of them relied on estimating a covariance matrix on each one of the data segments. We found this fact a critical issue because phonetic units tend to be too short in time, i.e. when dealing with small segments the estimation may lead to singular or badly estimate covariance matrices, in consequence this work has been developed using only the sum of square error (SSE) criterion.

Let $X = \{x_1, x_2, ..., x_n\}$ denote the sequence of feature vectors extracted from an utterance, where n is the length of X and each x_i is a d-dimensional feature vector. A segmentation that divides sequence X in k non overlapping contiguous segments can be denoted as $S = \{s_1, s_2, ..., s_k\}$, where $s_j = \{c_j, c_{j+1}, ..., e_j\}$ using c_j, e_j to represent the first and last indices of j^{th} segment. SSE criterion on S segmentation is defined as

$$SSE(X,S) = \sum_{j=1}^{k} \sum_{i=c_j}^{e_j} ||x_i - \widehat{m}_j||^2 \quad (1)$$

$$\widehat{m}_j = \frac{1}{(e_j - c_j)} \sum_{i=c_j}^{e_j} x_i \quad (2)$$

In the initial state, the algorithm defines one segment s_i for each x_i present in X, i.e., segments containing just one vector of the data. The algorithm iteratively merges segments

until it reaches the imposed number of segments. Given the segments s_j , s_{j+1} and R as the segment resulting from grouping of s_j and s_{j+1}, the grouping criterion is defined as:

$$\Delta SSE(X,j) = SSE(X,R) - SSE(X,s_j) - SSE(X,s_{j+1}) \quad (3)$$

The optimal grouping is performed by merging the adjacent segments such that the grouping criterion is the minimum. A direct application of the agglomerative clustering algorithm proposed by [8] leads to a high computational cost procedure. This is due to the extensive use of summation operations in grouping criterion. In this paper, we show a faster implementation of the original grouping criterion by means of re-writing the mathematics involved.

3. EFFICIENT CRITERIA

In order to efficiently implement the clustering step, we can pre-calculate a cumulative integral like function, helping to suppress redundant operations on the computation of equation 1.

$$G(i) = \sum_{k=2}^{i} x_{k-1} (G(1) = 0) \quad (4)$$

Using the previous function during SSE computation, it can be efficiently computed in terms of G function:

$$|s_j| = e_j - c_j + 1 \quad (5)$$

$$\widehat{m}_j = \frac{1}{|s_j|}(G(e_j+1) - G(cj)) \quad (6)$$

Despite this preprocessing speedup in the algorithm, we still found a major improvement that is not reported in [8]. The grouping evaluation criteria can be analytically reduced to a much faster computation [7] by using the following expression:

$$\Delta SSE = \sum_{i \in R} \|x_i - \widehat{m}_R\|^2 - \sum_{i \in s_j} \left\|x_i - \widehat{m}_j\right\|^2 - \sum_{i \in s_{j+1}} \left\|x_i - \widehat{m}_{j+1}\right\|^2$$

$$\Delta SSE = \sum_{i \in s_j} (\|x_i - \widehat{m}_R\|^2 - \left\|x_i - \widehat{m}_j\right\|^2) + \sum_{i \in s_{j+1}} (\|x_i - \widehat{m}_R\|^2 - \left\|x_i - \widehat{m}_{j+1}\right\|^2) \quad (7)$$

Using the following properties :

$$u^2 - v^2 = (u+b)\cdot(u-b) \quad (8)$$

$$\textstyle\sum_{i \in s_j} x_i = |s_j| \widehat{m}_j \quad (9)$$

$$\widehat{m}_R = \frac{|s_j|}{|s_j + s_{j+1}|}\widehat{m}_j + \frac{|s_{j+1}|}{|s_j + s_{j+1}|}\widehat{m}_{j+1} \quad (10)$$

Gives the following development:

$$\Delta SSE = \left[\sum_{i \in s_j} (x_i - \widehat{m}_R + x_i - \widehat{m}_j)\right] \cdot (\widehat{m}_j - \widehat{m}_R) + \left[\sum_{i \in s_{j+1}} (x_i - \widehat{m}_R + x_i - \widehat{m}_{j+1})\right] \cdot (\widehat{m}_{j+1} - \widehat{m}_R) \quad (11)$$

$$\Delta SSE = |s_j| |\widehat{m}_j - \widehat{m}_R||^2 + |s_{j+1}| |\widehat{m}_{j+1} - \widehat{m}_R||^2 \quad (12)$$

Yielding to this compact expression that has lost the sum operators:

$$\Delta SSE(X,j) = \frac{|s_j||s_{j+1}|}{(|s_j| + |s_{j+1}|)} ||\widehat{m}_j - \widehat{m}_{j+1}||^2 \quad (13)$$

This compact expression has lost the sum operators and consequently the resulting algorithm implementation can improve the computational cost substantially.

4. DIVISIVE CLUSTERING FOR OPTIMAL PHONETIC SEGMENTATION

Agglomerative hierarchical clustering (Bottom-up) starts defining one segment for each feature vector in the utterance and, at each iteration, merges the two closest segments. As ΔSSE criterion is oriented to variance minimization, the algorithm first merges smaller segments before starting to merge bigger ones. While segments are small, containing just 3 or 4 feature vectors, the estimation of its statistical properties is badly conditioned. Both these facts make the agglomerative nesting being badly conditioned during most of its iterations. Also, the number of iterations of Bottom-up approach is high because for each final segment bottom-up clustering has to perform as many iterations as vectors compose it. Previous observations motivated us to explore divisive clustering (Top-down) approach for segmentation and compare both. Initially in Divisive formulation, all the speech data forms a unique segment and at each step a divisive method splits up a cluster into two smaller ones, until it finally reaches k optimal segments. Each iteration divides a cluster, let us call it R, into two clusters A and B. The criterion to follow in order to find out where to split a segment is based on ΔSSE (shown in equation 3). In this way it can be shown that dividing the segment R into two segments; A and B, in terms of minimizing the sum of square error is equivalent to maximizing ΔSSE criterion show in algorithm 1 [7].

Algorithm 1: Divisive Segmentation Algorithm

Data: sequence $X = (x_1, .., x_n)$, number of segments k
Result: Segmentation $S = \{s_1, ..., s_k\}$ where $s_j = \{c_j, c_j + 1, .., e_j\}$ & $|s_j| = (e_j - c_j) + 1$
Initialization: $S = \{s_1 = \{1, .., n\}\}$;
while *number of segments in* $S < k$ **do**
  find indexes (i, j) such that maximizes following equation:
  Let $R = s_j$, $A = s_j\{cj, ..i\}$ and $B = s_j\{i+1, ..., e_j\}$.

$$\Delta SSE(X, s_j) = \frac{|A||B|}{|R|} ||\widehat{m}_A - \widehat{m}_B||^2 \quad (14)$$

  Update S by splitting s_j into segments A and B;
end

5. NUMBER OF SEGMENTS ESTIMATION

Divisive segmentation, as well as agglomerative previous work techniques [8], relies on prior knowledge of the number of segments. Many applications that require segmentation do not have prior knowledge of the number of segments in the data. Consequently this assumption is problematic and unpractical. Our contribution in this paper is to explore the utilization of a criterion in order to stop the clustering and implicitly estimate the number of segments from the data. This approach would overcome the problem of the manual number of segments selection, making the algorithms totally blind. The stopping criterion in the context of speech hierarchical clustering has been extensively studied in the Diarization literature. In general Diarization approaches use agglomerative hierarchical clustering for grouping together long speech segments (more than 1 second) belonging to the same speaker characteristics. Information change rate (ICR) [5] is a measure derived from a generalized likelihood ratio (GLR) which has been proposed as a robust method for automatically stopping the clustering in speaker segmentation tasks. It is designed in order to be able to compare segments without taking into account differences in size. We adopted ICR measure and applied it in order to characterize the phonetic segmentation status, allowing us to determine when it is adequate to stop the clustering. During hierarchical clustering, at iteration t the current segmentation, $S_t = \{s_i, ..., s_k\}$, is evaluated in terms of ICR measure using the following expression:

$$Stopcriteria_t = \frac{1}{k-1}\sum_{j=1}^{k-1} ICR(s_j, s_{j+1}) \tag{15}$$

Assuming Gaussian distributions, being Σ the covariance matrix:

$$H(a) = \frac{1}{2} ln(|\Sigma_a|) \tag{16}$$

$$ICR(s_j, s_{j+1}) = H(s_j \cup s_{j+1}) - \frac{(|s_j|H(s_j) + |s_{j+1}|H(s_{j+1}))}{(|s_j| + |s_{j+1}|)} \tag{17}$$

This criterion can be seen as a summarization of the degree of homogeneity between each pair of consecutive segments (sharing a boundary) resulting from the segmentation. The estimation of the number of segments is obtained by stopping the algorithm when segmentation in terms of ICR achieves a given threshold. We faced the problem of badly estimated covariances using covariance regularization: $\widehat{\Sigma_a} = I + \frac{\Sigma_a}{\alpha}$ using $\alpha = 0.05$. It must be noted that as covariance estimation is taking part only of the stopping criterion, regularization does not affect boundary estimation performance. In order to tune the stopping threshold under our spectral analysis framework, we used the TIMIT training set. Our objective was to analyze the relation between ground truth estimation and the stopping criteria. We used the 4200 phonetically annotated sentences in training set to compute the ICR measure on the ground truth manual segmentation. We analyzed the distribution of the ICR values from the sentences in training set, figure 3, to estimate the optimal threshold.

6. EXPERIMENTS

We performed the experiments on the TIMIT corpus [3] using the corpus training set in order to estimate the threshold for the stop criteria and the test set in order to validate the algorithms. For each sentence we computed the spectral features from speech signal using a pre-emphasis filter (factor was set to 0.97) and a 20 millisecond Hamming window with a 5 millisecond shift. The Mel Filtered spectrogram is generated by a reference software [4] using 50 triangular melodic scale filters between 133Hz and 8000Hz. This spectral framework was chosen as it was experimentally determined as optimal for our posterior processing steps [11]. Initially we performed the experiments comparing both clustering approaches: agglomerative and divisive segmentation, using a simplified scenario where we assumed prior knowledge of the number of segments (setting k as the number of phonemes in the utterance) . As previously stated, this is an unrealistic prior knowledge assumption. Consequently posteriorly we performed experiments using the blind approaches, that is using the ICR stopping criterion for which the threshold was set to 30.

6.1 Segmentation evaluation measures

Different measures have been reported in the literature in order to evaluate the segmentation task. Segmentation evaluation is usually based in measures related to precision and recall. For some finite section of speech let N_{hit} be the number of boundaries correctly detected and N_{ref} be the total number of boundaries in the reference. HR can then be calculated using equation 18. Another central measure, especially in the case of blind methods, is the over-segmentation (OS) rate, which exposes the relation between the total number of detected boundaries N_f and the number of boundaries in the reference N_{ref}.

$$HR = \frac{N_{hit}}{N_{ref}} \cdot 100 \tag{18}$$

$$OS = (\frac{N_f}{N_{ref}} - 1) \cdot 100 \tag{19}$$

Although hit rate (HR) and over-segmentation (OS) are mostly used, recent work proposes a combination of both into a reliable single-value measure: the R-value [9].

$$r1 = sqrt((100 - HR)^2 + OS^2) \tag{20}$$

$$r2 = \frac{HR - 100 - OS}{sqrt(2)} \tag{21}$$

$$R - value = 1 - \frac{abs(r1) + abs(r2)}{200} \tag{22}$$

The R-Value is especially interesting because it faces the problem of boundary matching. During boundary matching a tolerance collar (usually expressed in millisecond) is set around reference boundaries allowing matching with an hypothesized boundary falling inside this region. Most of the literature evaluation measures require this preprocessing step and strongly depend on it but do not specify clearly how boundary matching problems are solved. In [9] an analysis of the boundary matching potential problems are presented and

[4] Dan Ellis. Lab Rosa Matlab Audio Processing Examples

a solution is proposed by avoiding overlap in boundary tolerance collars. We applied this boundary matching method during the computation of the evaluation measures for showing our experimental results.

6.2 Hierarchical clustering Segmentation results

We present our results using two formats: detailed tables with the average values and figures to show distribution of the segmentation results over the test set. Comparison between Top-down and Bottom-up approaches are shown in table 1 and figure 4. As initial experiment used prior knowledge of number of segments, only recall rates are shown, being an equivalent to Hit Rate measure. Both approaches exposed very similar performance. However, Bottom-up approach achieves sightly higher results detecting correct boundaries.

The results for the blind approaches using the stopping criterion are shown in tables 2 and 3. Complementary, figure 5 shows the distribution of the R-value score for the validation data set. Again, both approaches exposed a very similar performance, with a high recall rate and low over segmentation. Overall, distribution of the results shows that the R-value is around 80 points depending on the tolerance allowed for the boundary matching procedure. Comparing the algorithms with previous approaches exposed in [9], we found that proposed blind Hierarchical clustering approaches obtain a sightly higher Hit Rate while obtaining a significant reduction in over segmentation. Concretely the closest approach in terms of performance [1] scored an average hit rate close to 80% with an average over segmentation of 10%.

Analyzing the obtained results we have observed that many of the segmentation errors (approximately 40% of the boundary detection errors) are boundaries placed in order to separate different speech patterns inside phoneme segments. As an example we include the blind segmentation of the word 'greasy' in figure 1 and the associated phonetic ground truth in figure 2. Note the wrongly placed extra boundary in the first segment, despite annotation is defining only one segment. Speech contained in the segment can be easily perceived as two different patterns, causing the algorithm to include a boundary between them and causing an error. These kind of insertions can also happen during phonetic segments exposing a rising or falling formant, segments annotated as silences including some noise sources like coughs and phonemes elements exposing non smooth behavior like heavy attack or small aspirations. In general, algorithms tend to capture all the structures inside speech, in consequence the number of phonemes in the ground truth is not an exact estimation of the real number of structures in the data. Structure boundaries compete with phonetic boundaries for detection, making that an abrupt and clear structures (as cough inside a silence segment) obtain a boundary instead of more subtle phonetic boundaries. In general, we determined experimentally that allowing a small percentage of oversegmentation would raise phonetic boundaries Hit Rate.

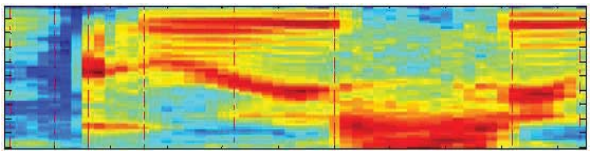

Figure 1: *Blind segmentation for the word greasy.*

Algorithm	agglomerative nesting	divisive approach
20 ms	0.7714	0.7653
30 ms	0.7995	0.7921
40 ms	0.8104	0.8021

Table 1: Recall rates for agglomerative versus divisive approaches

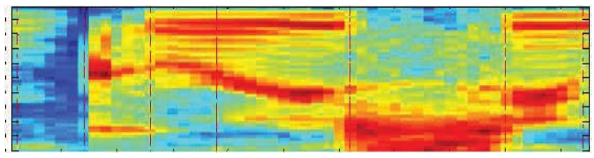

Figure 2: *Ground truth phonetic segmentation for the word greasy.*

Interval	HR	OS	Precision	Recall	R-value
20ms	74.335	-2.632	0.773	0.743	0.769
30ms	77.071	-2.632	0.801	0.771	0.790
40ms	78.138	-2.632	0.812	0.781	0.799

Table 2: Experimental results for Divisive segmentation (Top-down) using stopping criterion

Interval	HR	OS	Precision	Recall	R-value
20ms	75.592	-1.824	0.780	0.756	0.777
30ms	78.577	-1.824	0.810	0.786	0.800
40ms	79.763	-1.824	0.822	0.798	0.809

Table 3: Experimental results for Agglomerative-nesting (Bottom-up) using stopping criterion

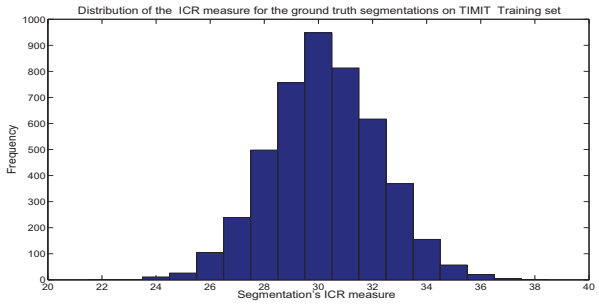

Figure 3: *ICR rates for real number of segments.*

7. CONCLUSIONS

In this paper we present an analysis of different strategies to phonetic segmentation using hierarchical clustering. Starting from previous research [8], we developed a Top-down formulation of the phonetic segmentation using divisive clustering and compared the properties of both Bottom-up and Top-down approaches. In addition, we expose an improvement for both algorithms implementation by the reformulation of the math involved. As we have seen, these approaches require the prior knowledge of the number of final segments. In

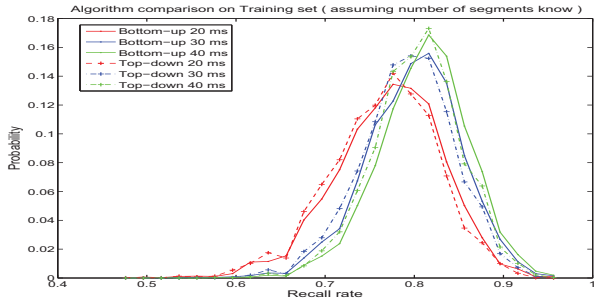

Figure 4: *Distribution of recall rates over Training dataset.*

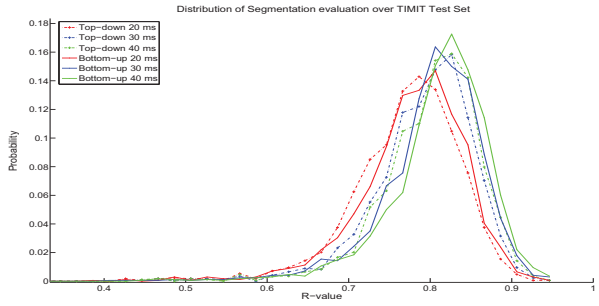

Figure 5: *R-value score distribution over Test dataset.*

consequence, an extension for both algorithms is presented in order to estimate the number of phonetic segments inside the data. The estimation process is based on a robust criterion making the algorithms suitable for being used in real applications where the number of phonemes inside the data is unknown. Our results show a high Hit Rate of approximately 80% and a low over-segmentation rate of approximately -1.8. Overall these results makes the algorithms suitable for being used in large video sequence, providing a fast response capable of work in real time.

8. ACKNOWLEDGMENTS

This work was partially funded by the Spanish MITC under the "Avanza" Project *Ontomedia* (TSI-020501-2008-131).

REFERENCES

[1] G. Aversano, A. Marinaro, et al. A new text-independent method for phoneme segmentation. In *Proceedings of the 44th IEEE 2001 Midwest Symposium on Circuits and Systems, 2001. MWSCAS 2001*, volume 2, pages 516–519, 2001.

[2] S. Dusan and L. Rabiner. On the relation between Maximum Spectral Transition Positions and phone boundaries. In *Ninth International Conference on Spoken Language Processing*, 2006.

[3] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, and N. Dahlgren. DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. *NTIS order number PB91-100354*, 1993.

[4] I. Gholampour and K. Nayebi. A New Fast Algorithm for Automatic Segmentation of Continuous Speech. In *Fifth International Conference on Spoken Language Processing*. ISCA, 1998.

[5] K. J. Han, S. Kim, and S. S. Narayanan. Strategies to improve the robustness of agglomerative hierarchical clustering under data source variation for speaker diarization. *IEEE Transactions on Audio, Speech & Language Processing*, 16:1590–1601, 2008.

[6] G. Hu and D. Wang. Auditory segmentation based on onset and offset analysis. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(2):396–405, 2007.

[7] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley & Sons, 1990.

[8] Y. Qiao, N. Shimomura, and N. Minematsu. Unsupervised optimal phoneme segmentation: Objectives, algorithm and comparisons. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 3989–3992, 2008.

[9] O. J. Räsänen, U. K. Laine, and T. Altosaar. An improved speech segmentation quality measure: the r-value. In *Interspeech*, pages 1851–1854, 2009.

[10] M. Sharma and R. Mammone. Blind speech segmentation: automatic segmentation of speech without linguistic knowledge. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 2, pages 1237–1240. IEEE, 2002.

[11] T. P. Tony Ezzat. Discriminative word-spotting using ordered spectro-temporal patch features. In *SAPA workshop*, 2008.