Improved performances and automatic parameter estimation for a context-independent speech segmentation algorithm

Guido Aversano[1,3] and Anna Esposito[2,3]

[1] Dipartimento di Fisica "E.R. Caianiello", Università di Salerno, Italy,
guido.aversano@sa.infn.it,
[2] Department of Computer Science and Engineering, Wright State University, Dayton, Ohio, USA,
anna@cs.wright.edu,
[3] International Institute for Advanced Scientific Studies (IIASS), Vietri sul Mare (SA), Italy

Abstract. In the framework of a recently introduced algorithm for speech phoneme segmentation, a novel strategy has been elaborated for comparing different speech encoding methods and for finding parameters which are optimal to the algorithm. The automatic procedure that implements this strategy allows to improve previously declared performances and poses the basis for a more accurate comparison between the investigated segmentation system and other segmentation methods proposed in literature.

1 Introduction

The computational treatment of raw data, originated by real-word processes, usually requires a preliminary step in which the data has to be encoded in a form that is suitable for further processing. Ideally, the encoded data should retain only the portion of informational content which is useful to the particular task the machine is going to perform, whilst every useless information should be discarded. The choice of the encoding scheme can strongly influence the quality of the output for the overall computation. This is a well-known issue, at least from a theoretical point of view, and it is often addressed in literature as the "data preprocessing problem" (see, e.g., [1]).

Unfortunately, in the practice, there are frequent cases for which the most suitable way of encoding scheme is not known *a priori* and, therefore, it becomes necessary to test several different processing methods, making a comparative choice between them. A more complete description on this topic can be found in [2].

From an conceptual point of view, it appears evident that a consistent choice between encoding schemes cannot be performed without having previously defined a judging rule which is not ambiguous. This is a non-trivial task in many practical applications. An "on-the-field" instance , in which such difficulties are encountered and subsequently overcome, is presented in this paper.

The framework for what is going to be exposed is a recently introduced algorithm which performs the segmentation of speech into phonemes. A novel strategy will be used for comparing different types of encoding of the speech signal, in order to individuate which one best fits the algorithm and leads to a minimum segmentation error. Full details about the segmentation algorithm are given in the paper [3]; in the next section only a brief description of the segmentation task will be provided, in addition to a few formal definitions, which serve to numerically express the performance of the whole segmentation system. This is an essential background to the discussion carried out in sections 3 and 4, which are devoted to describe and to solve, respectively, the ambiguities encountered in comparing performances for this particular application.

2 The segmentation algorithm and measures of its performance

The investigated algorithm operates on encoded speech signal and tries to detect the exact position of the boundaries between phonemes. It is worth mentioning that the only constraint, imposed by the algorithm to the form of the speech encoding, is that it must be a time-sequence of vectors; so every "short-time" representation of the signal (i.e. any vector encoding a small time interval or "frame") can be used. The action of the algorithm is regulated by three parameters, namely a, b and c. The a and c parameters are integers, representing the number of speech frames taken into account in different phases of the algorithm implementation, whereas b takes values in the real domain. The b parameter can be pictorially described as a particular threshold level used to reject "candidates" to the role of phonemic boundary.

Shortly it will be shown that a fine tuning of the above parameters is not only desirable for granting optimal performance to the end-user, but it is also necessary for taking important decisions about the choice of the encoding technique underlying the entire segmentation process. The first step in this direction is defining some indices to quantify the quality of the performed segmentation, for given a, b and c, on a specified data set encoded using a particular processing technique. To this purpose, a collection of 480 sentences was extracted from the American-English DARPA-TIMIT database. These sentences are pronounced by 48 different speakers (24 females and 24 males). Each waveform of the DARPA-TIMIT has an associated labeling file, which contains the "true segmentation", i.e. the actual positions (in samples) of the phoneme boundaries manually detected by an expert phonetician.

Once the algorithm has performed its own segmentation, this is compared with the true segmentation of any sentence in the dataset: a phoneme boundary identified by the algorithm is defined as "correct" if it is placed within a range of $\pm 20\ ms$ (± 320 samples) from a true segmentation point. In figure 1 an example of speech waveform, taken from the database, is presented, together with the associated "true" and detected segmentation points.

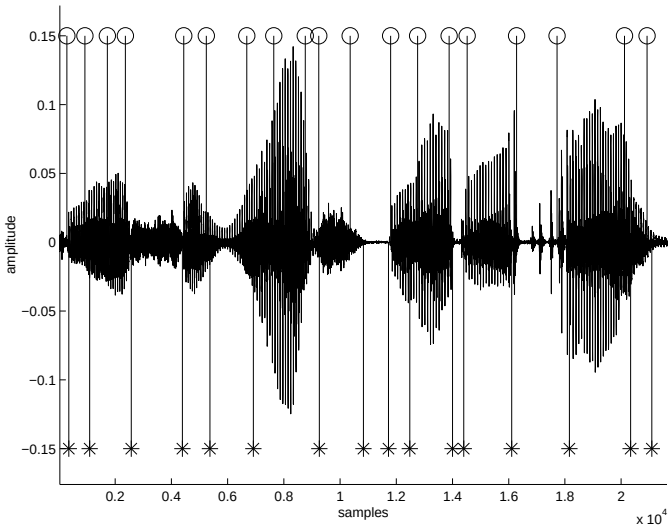

Fig. 1. *Speech waveform taken from database; "true" segmentation points are indicated by ∗; those detected by the algorithm by ∘.*

An index for expressing the algorithm's performance can be defined as the percentage of correctly detected phoneme boundaries:

$$P_c = 100 \cdot \frac{S_c}{S_t}, \tag{1}$$

where S_t is the total number of "true" segmentation points (S_t) contained in our database (17,930), and S_c the number of correctly detected points.

It is easy to prove that such index alone does not measure the quality of the performed segmentation. In fact, the algorithm could output a huge number of detected boundaries, incrementing, fictitiously, the probability of detecting true segmentation points, with the drawback of having introduced a large number of unwanted extra points (i.e. erroneous segmentation points). This phenomenon is known as *over-segmentation* and can be quantified by an over-segmentation rate D, defined as the difference between the total number of segmentation points detected by the algorithm (S_d) and S_t:

$$D = S_d - S_t. \tag{2}$$

An alternative measure of over-segmentation expressing the percentage of extra points, $D' = 100 \cdot (S_d/S_t - 1)$, can be found in Petek *et al.* (1996) [4].

3 The problem of optimizing and comparing performances

From what stated in the previous section, it clearly follows that the goal of an optimization procedure for the segmentation system should consist in maximiz-

ing the correct detection rate without letting the over-segmentation grow. The utility of such a procedure and the necessary requirements for its functioning will become clear after having examined the following experimental results on the already mentioned data set. Values for P_c and D, found with fixed a, b, c, and adopting various encoding schemes, are reported in Table 1. References for all the tested encoding methods are given in section 5.

Table 1. *Correct detection percentage and over-segmentation rate, found for different encoding schemes and fixed parameters $a = 2$, $b = 0.2$, $c = 6$.*

Encoding	P_c	D
5-PCBF	71.4%	459
8-PCBF	77.2%	2577
5-MelBank	74.4%	-606
8-MelBank	79.3%	1581

From the above results, it is not straightforward to decide which one, between 5-PCBF[1] and 8-PCBF encoding methods produced better performances. As a matter of fact, 8-PCBF gives a higher correct detection percentage than 5-PCBF, but it also resulted in a higher number of inserted extra-points. The same reasoning holds for the couple 5-MelBank and 8-MelBank. Notice the negative value for D in correspondence of the 5-MelBank encoding: there are also cases in which the number of points inserted by the algorithm is less than the number of effective segmentation points contained in the database.

Furthermore, the optimality of a particular triple of a, b, c parameters cannot be caught at a glance, due to the interdependency of the indices P_c and D (as noted before, a higher over-segmentation rate corresponds to a higher probability of correct detection). Therefore, their dependence from the free parameters is not just straightforward, as can be seen in Table 2, where it is shown how a variation in the parameter c influences both P_c and D, in a way that does not unveil the optimal value for c.

4 The "optimize-and-compare" procedure

It would be easy to compare the results reported in the above tables if all their rows showed the same numeric value for one of the two indices P_c or D, so that the performance evaluation could be based on the other index, taking the common value as a reference level. A situation of this kind could be experimentally induced by making several trials, in which two of the three parameters are fixed and the other one is "moved around", independently until a configuration is found where D, for example, has the same value for every encoding

[1] PCBF stands for "Perceptual Critical Band Features". See the last section for more details.

Table 2. *Correct detection percentage and over-segmentation rate for the 8-MelBank encoding scheme, obtained fixing $a = 2$, $b = 0.2$, and for three different values of the parameter c.*

c	P_c	D
4	80.8%	2564
5	80.1%	2048
6	79.3%	1581

scheme to be compared, and P_c remains the sole significant index for judging the performances of both the algorithm and the encoding scheme.

Such an approach, however, has at least two weak points. The first one is that the number of trials needed to find the desired equality for D is essentially a question of luckiness; even a skilled experimenter will be engaged in a time-consuming loop which consists in hypothesizing the value for the parameter subject to variation, waiting for the results of the relative experiment, correcting the hypothesis in the light of these results, making a new trial, and so on. What makes this process worse is that it should be repeated for every encoding scheme under testing, and for every couple of values that can be taken by the two remaining parameters. The other problem is related to the choice of the reference level: it would be better, for instance, if the common value chosen for D had some particular properties which can justify its adoption.

To overcome the first of the above limitations we implemented an automatic procedure that finds – given a particular encoding scheme and having fixed the value of the a and c parameters – the value of the parameter b, which satisfies the condition $D = 0$ after the execution of the segmentation algorithm. The choice of b as the "mobile" parameter and of $D = 0$ as the reference level can be motivated as follows.

It was already said that b plays the role of a threshold in the segmentation algorithm. Actually b, which is a real number falling within the interval $[0, 1]$, regulates almost directly the amount of segmentation points placed by the algorithm and consequently the over-segmentation D. Furthermore, having fixed $a = a^*$ and $c = c^*$, D as a function of b, $D(b) = D(a^*, b, c^*)$, is decreasing monotonic; when $b = 1$, D reaches its minimum value, $D = -S_t$, which is a negative number. The opposite extreme, $b = 0$, corresponds to eliminating the preliminary thresholding from the algorithm; in that point D assumes a maximum value which is not fixed but depends on a^*, c^* and on the chosen encoding. $D(0)$ is supposed to be greater or equal to zero.[2] Based on the above considera-

[2] Having $D(0) < 0$ would mean that the system, for every b, would always insert a number of segmentation points smaller than the number of effective segmentation points. Such a situation of "unreversible under-segmentation" would immediately suggest to change the adopted encoding scheme or the couple (a^*, c^*). Also note that $D(0) = 0$ is a very lucky circumstance, for which there would be no need to introduce thresholding in the algorithm.

tions, the choice of $D(b) = 0$ as constraint for the optimization of P_c appears the most natural one. An additional support to this choice comes from the fact that for text-dependent speech segmentation algorithms (i.e. those which rely on an externally supplied transcription for identifying phoneme boundaries) D equals 0 by definition [5].

The mentioned automatic procedure can be schematized by the following ten steps:

1. set a and c to some integer values, a^* and c^*, using an external control mechanism (see the last step);
2. run a few experiments, on the whole data set, using different values of b belonging to the interval $[0, 1)$;
3. evaluate the indices D and P_c for such experiments obtaining a set of sampling points for the functions $D(b)$ and $P_c(b)$;
4. identify a model function $\tilde{D}(b)$ (e.g. a polynomial) which fits the obtained sampling points to approximate the behavior of the function $D(b)$ (The choice of the model function is essentially dictated by empirical considerations on the distribution of the sampling points);
5. compute the zeros of $\tilde{D}(b)$; if the number of such zeros is greater than one, then the zero of interest will be the one that is found in the function's decreasing monotonic region to which the sampling points belong (e.g. if the model function $\tilde{D}(b)$ is a parabola only the zero associated to the descending branch should be considered, since the function we want to approximate, $D(b)$, is a decreasing monotonic function). The output of this step is an estimate value for b, $\bar{b}$, for which $D(\bar{b}) \simeq 0$;
6. identify, as for D, a model function $\tilde{P}_c(b)$ to approximate the behavior of the function $P_c(b)$;
7. estimate the detection rate as $\bar{P}_c^* = \tilde{P}_c(\bar{b})$;
8. run again the segmentation algorithm using as threshold $b = \bar{b}$;
9. if $D(\bar{b}) \neq 0$ then go to step 4, to get a new estimate using the additional sample point represented by $D(\bar{b})$ and $P_c(\bar{b})$; otherwise $P_c^* = P_c(\bar{b})$ is assumed to be the correct detection rate corresponding to a zero over-segmentation value;
10. $P_c(a^*, c^*) = P_c^*$ is returned to the external mechanism which cares for finding the maximum of the discrete function $P_c(a, c)$. This control routine will eventually restart the whole procedure using a new (a^*, c^*) couple.

The external mechanism introduced in the last step can be a simple schedule which executes the procedure a few times: experience showed that, for all the encoding schemes considered up to now, the values of a and c which maximize $P_c(a, c)$ and make D equal to zero are always found limiting their search within the intervals $a \in \{1, 2, 3\}$, $b \in \{4, 5, 6, 7\}$.

Figure 2 graphically shows how the whole method works: several values for the parameter c are compared, evidencing a maximum for $c = 7$.

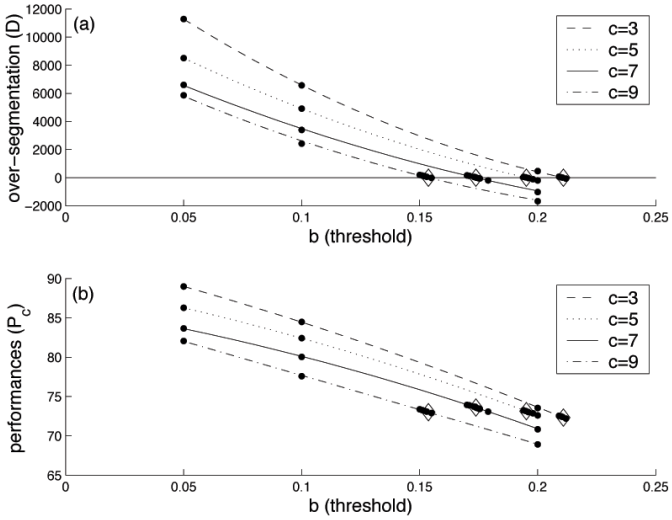

Fig. 2. *A graphical representation of the "optimize-and-compare" procedure, where four values of the parameter c are compared. The maximum correct detection rate P_c is encountered for $c = 7$.*

5 Preliminary results

The above procedure was embedded in the speech segmentation algorithms and tested on several speech encoding schemes, among which Mel-frequency Cepstral Coefficients (MFCC) [6], LPC [7], PLP [8], RASTA-PLP [9], Perceptual Critical Band Features (PCBF) and Mel-frequency Bank of filters (MelBank). The optimal segmentation results were obtained using the last two cited encodings; these can be both described as the output of a bank of filters, which span the whole frequency-range or a part of it. The subdivision of the frequency axis is not uniform and tries to reproduce the particular spectral resolution of the human ear. In addition, PCBF incorporates some other perceptual-based modification of the spectrum, i.e. loudness pre-emphasis and intensity-loudness compression. Note that the MelBank analysis is preliminary to the extraction of Mel-frequency Cepstral Coefficients, so its description is included in [6]. In the same manner PCBF analysis precedes PLP analysis [8].

The proposed automatic procedure made also easier to observe how the results varied when changing the number of PCBF and MelBank filters. The number of filters is indicated by the numeral preceding the name of the encoding method (e.g. 3-PCBF, 5-MelBank). Among all the tested encoding schemes the maximum percentage of correct detection was $P_c^* = 76.4\%$, obtained using the 8-MelBank encoding, and $a = 2$, $b = 0.23293$, $c = 6$ as values of the free parameters. The previously declared detection rate for the algorithm was 73.6% (using PCBF: see [3] for details), so a performance improvement of about 3% was realized just changing the encoding scheme.

6 Conclusions

Performance evaluation for speech segmentation methods is not straightforward. The interdependence between the various indices, which are usually used to express the quality of a performed segmentation, must be analyzed and exploited to formulate unambiguous rules for consistently comparing different methods and architectures. The present paper tries to give an answer to this issue, introducing an original methodology for choosing the optimal speech data encoding for a particular segmentation algorithm. Optimal tuning of the parameters that regulate the algorithm is also feasible, using the fully automated procedure proposed above. Further works will include the engineering of a technique, based on the same procedure, for finding optimal parameters from a small subset of data. A detailed comparison, supported by performance evaluations, between the investigated segmentation system and other methods proposed in literature is also underway.

Acknowledgements

The authors would like to thank Prof. Maria Marinaro and Antonietta Esposito for useful suggestions and collaboration. This work has been supported by the NSF KDI program, Grant No. BCS-9980054 "Cross-modal analysis of speech signal and sense: multimedia corpora and tools for gesture, speech, and gaze research" and by NSF Grant No. 9906340 "Speech driven facial animation".

References

1. Bishop, C. M.: *Neural Networks for Pattern Recognition*. Clarendon Press (1995)
2. Esposito, A.: *The importance of data for training intelligent devices*. "From Synapses to Rules: Discovering Symbolic Rules from Neural Processed Data", proc. of 5^{th} International School of Neural Nets "E.R. Caianiello", Apolloni B. and Kurfus K. (eds), Kluwer Academic Press (to appear)
3. Aversano, G., Esposito, A., Esposito, A., Marinaro, M.: *A New Text-Independent Method for Phoneme Segmentation*. Proc. of 44th IEEE Midwest Symposium on Circuits and Systems **2** (2001) 516–519
4. Petek, B., Andersen, O., Dalsgaard, P.: *On the robust automatic segmentation of spontaneous speech*. Proc. of ICSLP '96 (1996) 913–916
5. Pellom, B. L., Hansen, J. H. L.: *Automatic segmentation of speech recorded in unknown noisy channel characteristics*. Speech Communication **25** (1998) 97–116
6. Duttweiler, D., Messerschmitt, D.: *Nearly instantaneous companding for nonuniformly quantized PCM*. IEEE Transactions on Communications **COM-24** (1976) 864–873
7. Rabiner, L., Juang, B.: *Fundamentals of Speech Recognition*. Prentice-Hall (1993)
8. Hermansky, H.: *Perceptual Linear Predictive (PLP) Analysis of Speech*. Jour. Acoust. Soc. Am. **87(4)** (1990) 1738–1752
9. Hermansky, H., Morgan, N.: *RASTA Processing of Speech*. IEEE Trans. On Speech and Audio Processing **2(4)** (1994) 578–589