Albayzin 2014 Evaluation: TES-UAB Audio Segmentation System

Héctor Delgado and Javier Serrano

CAIAC, Autonomous University of Barcelona, Spain
{hector.delgado,javier.serrano}@uab.cat

Abstract. This paper describes the audio segmentation system developed by Transmedia Catalonia / Telecommunication and Systems Engineering Department, at the Autonomous University of Barcelona (UAB), for the Albayzin 2014 Audio Segmentation Evaluation. The evaluation task consists in segmenting spoken audio documents into three different acoustic classes (speech, background noise, and music), taking into account that more than one class may be present at any given time instant. Furthermore, additional difficulty has been added by fusing and merging audio from different databases. The proposed system is based on the recently presented "Binary Key" modeling approach, originally developed for speaker recognition, but successfully applied to other pattern recognition tasks, such as speaker diarization, emotion recognition and speech activity detection. Experiments carried out on the provided development data show a Segmentation Error Rate of 22.71%.

Keywords: audio segmentation, binary key, binary key background model

1 Introduction

Audio segmentation is the task of detecting the boundaries between different acoustic sources or classes within an audio signal. Over the years, audio segmentation has become an important task as a pre-processing tool for subsequent speech related tasks, such as Automatic Speech Recognition (ASR), speaker diarization, or Spoken Document Retrieval (SDR). Accurate audio segmentation labels are required to assure success of further systems.

In the last three editions of the "Jornadas en Tecnologías del Habla", audio segmentation evaluations have been conducted in the ambit of the Albayzin Evaluations. These evaluations aim at promoting research in the field of audio and speech processing, including audio segmentation, speaker diarization, language recognition and search on speech. With regard to audio segmentation, past evaluations have shown that the challenge is still far from being completely solved.

Recently, a speaker modeling technique called "binary key" was introduced in [4]. The approach provides a compact representation of a speaker model through a binary vector (vector only containing zeros and ones) by transforming the continuous acoustic space into a discrete binary one. The technique has also

been successfully applied to speaker diarization [5], emotion recognition [6], and Speech Activity Detection (SAD) [7]. This latter work is specially interesting in this ambit since it proposes a novel SAD approach achieving state-of-the-art performance. In fact, SAD can be considered as a particular case of audio segmentation, where speech and nonspeech acoustic classes are considered. Then, it may seem reasonable to think that this SAD approach may be useful for audio segmentation tasks involving more audio classes, such as speech, music, background noise, and combinations of all of them. Following these thoughts, an audio segmentation system based on binary keys has been developed to be evaluated in the Albayzin audio segmentation evaluation.

The paper is structured as follows: Section 2 gives an overview of the Albayzin 2014 audio segmentation evaluation. Section 3 describes the audio segmentation system based on binary keys. Section 4 describes the experimental setup and results. Section 5 concludes and proposes future work.

2 Audio segmentation evaluation

This section briefly describes the Albayzin 2014 audio segmentation evaluation (refer to [10] for an in-depth description).

As in the 2012 Audio Segmentation Evaluation, the task consists in segmenting a set of broadcast audio documents into segments according to a series of audio classes. These classes are speech, music, and noise. However, combinations of the three classes can occur in the audio being evaluated (overlapping classes). Therefore, a multiple layer labeling must be provided by the segmentation system.

For this evaluation campaign, the main change is related to the audio data to be processed. The test data consist of audio from different merged, or even overlapped, databases. This modification drastically increases the difficulty of the task and has as main aim to test the robustness of systems across different acoustic conditions.

2.1 Database description

The database proposed for this evaluation is a combination and fusion of three databases.

The first database is a broadcast news database from the 3/24 TV channel. The database was recorded under the Tecnoparla project [2] and contains around 87 hours of recordings.

The second dataset is the Aragón Radio database from the Corporación Aragonesa de Radio y Televisión, which provided the data for the Albayzin 2012 evaluation.

The third database is composed of environmental sounds from Freesound.org [1] and HuCorpus [9] among others. These sounds are merged with segments from the two previous databases.

All the data are supplied in PCM format, 1-channel, little endian 16 bit-per-sample, 16 KHz sampling rate.

2.2 Segmentation scoring

To evaluate systems, the Segmentation Error Rate (SER) is computed as the fraction of correctly attributed class time. This score is computed over the entire file to be processed, including regions containing overlapped classes. The metric is calculated as the Diarization Error Rate (DER) proposed in the NIST RT Evaluations [3].

Given a test dataset Ω, each document is divided into contiguous segments at all class change points. Then, the segmentation error time Ξ is computed for each segment n as

$$\Xi(n) = T(n)[\max(N_{ref}(n), N_{sys}(n)) - N_{correct}(n)] \quad (1)$$

where $T(n)$ is the duration of segment n, $N_{ref}(N)$ is the number of reference classes that are present in segment n, and $N_{Correct}(n)$ is the number of reference classes in segment n correctly assigned by the segmentation system. Then, SER is calculated as

$$SER = \frac{\sum_{n\in\Omega} \Xi(n)}{\sum_{n\in\Omega}(T(n)N_{ref}(n))} \quad (2)$$

SER includes three types of error, namely the Class Error Time, the Missed Class Time, and the False Alarm Class Time. The Class Error Time refers to the amount of time which has been assigned to an incorrect class. The Missed Class Time is the amount of time that a class is present but not labeled by the system. And the False Alarm Class Time refers to the amount of time which has been assigned to a class that is not present in the reference.

In order to take into account possible uncertainty and reference inconsistencies due to human annotations, a forgiveness collar of 1 second is applied to all reference boundaries.

3 Audio segmentation system description

The proposed audio segmentation system is inspired in the SAD system developed in [7], and adapted to the needs of the audio segmentation task of this evaluation.

The binary key modeling aims at transforming the input acoustic data into a binary representation, called binary key, which contains class-specific information, and therefore it is useful for discriminating between acoustic classes. This transformation is done thanks to a UBM-like model called Binary Key Background Model (KBM). Once the binary representation of the input audio is obtained, subsequent operations are performed in the binary domain, and calculations mainly involve bit-wise operations between pairs of binary keys.

3.1 KBM training

In this paper, the KBM is trained as follows (figure 1): First, a GMM is trained for each acoustic class (e.g., “speech”, “noise”, “music”) using Expectation-Maximization (EM) algorithm with appropriate labeled training data. Then,

the final KBM is the result of pooling all Gaussian components of the individual GMMs together. As an example, a KBM build from three classes 16-component GMMs will contain 32 Gaussian components in total.

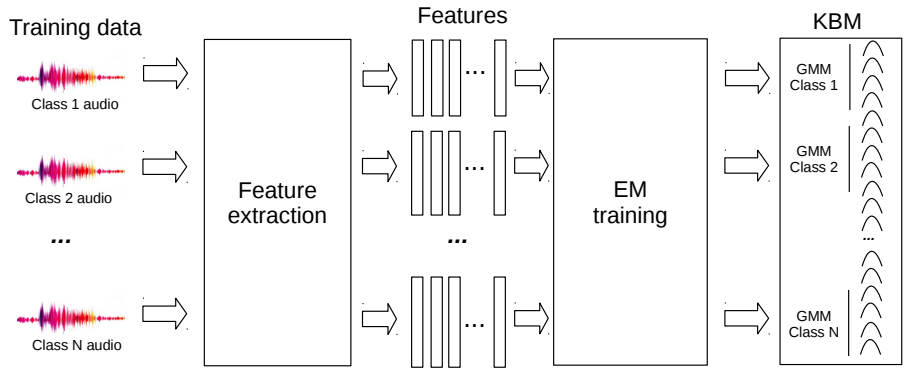

Fig. 1. KBM training process.

3.2 Binary Key computation

Once the KBM is obtained, any set or sequence of acoustic feature vectors can be converted into a Binary Key (BK). A BK $\mathrm{v}_f = \{v_f[1], ..., v_f[N]\}, v_f[i] = \{0, 1\}$ is a binary vector whose dimension N is the number of components in the KBM. Setting a position $v_f[i]$ to 1 (TRUE) indicates that the ith Gaussian of the KBM coexists in the same area of the acoustic space as the acoustic data being modeled. The BK can be obtained in two steps. Firstly, for each feature vector, the best N_G matching Gaussians in the KBM are selected (i.e., the N_G Gaussians which provide highest likelihood for the given feature), and their identifiers are stored. Secondly, for each component, the count of how many times it has been selected as a top component along all the features is calculated, conforming a Cumulative Vector (CV). Then, the final BK is obtained by setting to 1 the positions of the CV corresponding to the top M Gaussians at the whole feature set level, (i.e., the Mth most selected components for the given feature set). Note that this method can be applied to any set of features, either a sequence of features from a short audio segment, or a feature set corresponding to a whole acoustic class cluster.

3.3 Audio segmentation process

The audio segmentation process is illustrated in figure 2. First of all, the input feature vectors must be converted to a sequence of binary keys. The input data are divided into fixed-length segments, considering some overlap and window

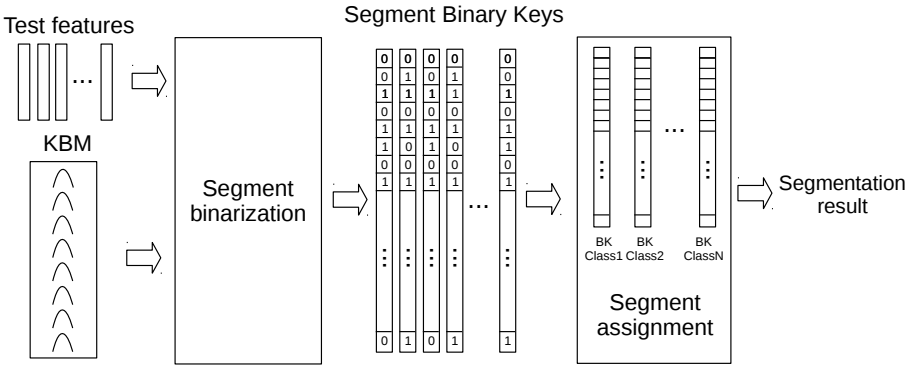

Fig. 2. Segmentation process

rate. Then, a BK is obtained for each segment by using the KBM following the method explained in section 3.2. From here on, all operations are performed in the binary domain. Segment assignment is done by comparing each segment BK with the N BKs (previously estimated using the KBM and training data) for each of the N target audio classes. Finally, the current segment is assigned to the class which maximizes the similarity between the BKs pair. The similarity between two binary keys a and b, according to [5] is computed as

$$S(\mathrm{a},\mathrm{b}) = \frac{\sum_{i=1}^{N}(a[i] \wedge b[i])}{\sum_{i=1}^{N}(a[i] \vee b[i])} \tag{3}$$

where $\wedge$ indicates the boolean AND operator, and $\vee$ indicates the boolean OR operator. This is a very fast, bit-wise operation between two binary vectors.

In addition, alternatives to the similarity calculation involving CVs are also tested in this work. First, the Intersection and Symmetric Differences Similarity, proposed in [8], is defined as

$$S_{ISDS}(\mathrm{A},\mathrm{B}) = \frac{\sum_{i=1}^{|A\cap B|} a_i + b_i}{(\sum_{j=1}^{A-B} a_j + \sum_{j=1}^{B-A} b_j) * \sum_{i=1}^{|A\cap B|} |a_i - b_i|} \tag{4}$$

where $\{\forall a \in A, \forall b \in B | A - B \neq \emptyset, \exists a \neq b | (a,b) \in A \cap B\}$. Here, the binary vectors act as indexes for the calculations with the cumulative vectors.

Finally, a simple cosine similarity between CVs is tested as well:

$$S_{cos}(a,b) = \frac{a \cdot b}{\|a\| \, \|b\|} \tag{5}$$

where a and b are the CVs being compared.

4 Experiments and results

As in the 2012 audio segmentation evaluation, a multi-layer labeling is requested when overlapped classes are present. However, in this system all possible combinations of the three proposed classes (speech, noise, and music) are taken as

separated classes as a starting point. At the end, the obtained segmentation is post-processed in order to get the final multi-layer labeling.

This year, the UAB group is submitting two different systems. They mainly differ in the way the KBM is obtained, keeping the rest of the setting unaltered for both systems. These common settings are explained next.

First, the provided database, which consists of 20 audio excerpts of around 1 hour each one, is divided into two subsets. The first one is composed of the first 14 audio files (around 70% of the corpus) and it is used for training. The rest (6 audio files conforming the remaining 30%) is used for testing.

For feature extraction, LFCCs are extracted from the audio signal using a 20ms analysis frame, a shift of 10ms, and a Hamming window. 12 static coefficients are extracted plus the energy coefficient, delta, and delta-delta coefficients, totaling 39 coefficients. The tool used for feature extraction is the SPro toolkit (`https://gforge.inria.fr/projects/spro/`).

Regarding binary key computation, the top 5 Gaussian components are taken in a frame basis. Several values of the factor of top Gaussians at segment level M are tested in the experiments (0.1 and 0.15).

Finally, in the data assignment stage, binary keys are computed for each 0.3s segment, augmenting it 2.5s before and after, totaling 5.3s. This is done in order to have sufficient data to estimate the BKs, but also for avoiding very over-segmented labels. Then, the window is shifted 0.3s to calculate the next BK.

4.1 Primary and contrastive systems

As said above, the two systems share a common setting, but differ in the way the KBM is obtained.

In the primary system, all combinations of the three proposed acoustic classes are considered, totaling 7 combinations. Therefore, seven GMMs are trained ("sp", "no", "mu", "sp+no", "sp+mu", "sp+no+mu", "no+mu"), and the final KBM is the result of pooling all Gaussian components. However, in the contrastive system, only the three proposed classes are considered (speech, noise, and music). Therefore, in this case three GMMs are trained.

After training the KBM, in both systems BKs are estimated for the 7 combinations, resulting in 7 BKs which act as acoustic models for each class combination. Note that this is done for both systems, regardless of the number of classes used to conform the KBM. In order to clarify this, table 1 summarizes the number of components of KBM depending on the number of classes and the number of individual GMM components.

4.2 Experimental results and discussion

Table 2 and table 3 show the SER of the primary and contrastive systems, respectively, for different KBM sizes, different values of M, and the different proposed similarity measures, evaluated on the test dataset (note that this test dataset

Table 1. Number of components of KBM depending on the number of classes being considered.

Primary system (7 classes)		Constrastive system (3 classes)	
Components per class	KBM components	Components per class	KBM components
128	896	128	384
256	1792	256	768
512	3584	512	1536
1024	7168	1024	3072

is extracted from the development files provided, as the official test ground-truth segmentation labels of the evaluation were not available at the moment of writing this paper). The best performing configuration of the primary system comprises a 3584-component KBM (i.e. 512 Gaussians per class combination), $M = 0.1$, and using the cosine similarity, providing an overall SER of 22.71%. The rest of configurations performances oscillate between 23% and 28% SER in the primary system, and between 26% and 30% in the contrastive system. It is also observed that the choice of similarity measure has more impact in performance that the value of M. The best performing similarity measure is the cosine similarity, followed by the ISDS similarity and the similarity given by equation 3.

It also can be seen that the primary system outperforms the contrastive one, even using a lower number of Gaussian components.

Table 2. SER of primary system on the test dataset, according to the number of KBM components, the factor M of top Gaussians per segment, and the used similarity metric. Best results for each similarity measure are highlighted.

SER of primary system (%)					
KBM components	S		S_{ISDS}		S_{cos}
	$M = 0.1$	$M = 0.15$	$M = 0.1$	$M = 0.15$	-
896	28.87	28.23	26.13	26.52	24.29
1792	28.73	28.49	25.58	26.72	23.76
3584	28.28	**26.17**	**24.71**	25.66	**22.71**

Table 4 gives individual results for each audio file with the best-performing configuration of the primary system, by breaking down SER into Miss Class Time, False Alarm Class Time, and Class Error Time. In general, miss errors become the most contributing part of the total error, with rates between 9.5% and 12.3%, and an overall rate of 10.4%. False alarm errors are lower than miss errors, but quite high for some audio files (up to 12.8%), totaling an overall rate of 6.8%. Finally, class errors are also lower than miss errors, and slightly lower than false alarm errors, with values oscillating between 2.8% and 8.5%.

Table 3. SER of contrastive system on the test dataset, according to the number of KBM components, the factor M of top Gaussians per segment, and the used similarity metric. Best results for each similarity measure are highlighted.

SER of contrastive system (%)					
KBM components	S		S_{ISDS}		S_{cos}
	$M = 0.1$	$M = 0.15$	$M = 0.1$	$M = 0.15$	-
768	29.62	30.08	27.49	28.5	26.73
1536	28.32	28.35	**26.86**	27.13	**25.0**
3072	**28.04**	28.7	27.2	27.73	25.64

Table 4. Most accurate system results per audio file, broken-down into error types: Miss Class Time (Miss), False Alarm Class Time (FA), Class Error Time (Class), and Segmentation Error Rate (SER).

File ID	Miss	FA	Class	SER
track15	10.0	9.1	3.0	22.14
track16	10.3	6.0	7.0	23.39
track17	10.0	4.3	2.8	17.12
track18	12.3	4.4	7.2	23.7
track19	10.1	5.2	5.2	20.47
track20	9.5	12.2	8.5	30.21
Overall	10.4	6.8	5.6	22.71

After analyzing results of primary and contrastive systems on development data, the best performing parameter settings are taken to be used to process the official evaluation test dataset. The setting for the primary system is 3584 KBM components and cosine distance. Regarding the contrastive system, 1536 KBM components and the cosine distance are selected.

By using the selected settings, the test dataset is then processed. The system in which audio segmentation was performed is a Debian Wheezy virtual machine with 12 assigned GB RAM, running on an Intel Xeon E5-2420 at 1.90GHz CPU. Table 5 shows execution time and real time factor (xRT) for both primary and contrastive systems. During the experiments, it has been observed that the most time consuming part of the segmentation systems is the log-likelihood computation of all the input features for each Gaussian components, needed to estimate the binary keys. This stage is speeded up significantly when the KBM size decreases. After this stage, data assignment is a very fast stage.

Table 5. CPU time (hh:mm:ss) and Real Time Factor (xRT) of primary and contrastive systems on the official test data (total time is 15:37:43).

	Primary system		Constrastive system	
Task	Time	xRT	Time	xRT
Feature extraction	00:02:17	0.002	00:02:17	0.002
Audio segmentation	07:13:02	0.462	03:11:38	0.204
Overall	07:15:20	0.464	03:13:55	0.207

Primary system presents an overall xRT of 0.464. Although faster than real-time, this execution time could be too long for some time-critical applications. Contrastive system shows a xRT of 0.207, which is significantly lower than the primary system (more than twice faster). Although experimental results have shown weaker performance than the primary system, the contrastive system could be useful when higher speed is required, at the cost of a slight decrease of accuracy.

5 Conclusions

An audio segmentation system based on binary key modeling has been developed an submitted to the Albayzin 2014 audio segmentation evaluation. The system performs audio segmentation by annotating the input data according to all possible combinations of the three proposed audio classes, and finally the obtained labels are post-processed in order to get the final, multi-layer labeling. The proposed approach is based on the Binary Key modeling, and has been tested with a primary system and a contrastive system. Those systems only differ in the way the KBM is trained by considering only the three classes or all possible combinations when training GMMs to conform the KBM.

Experiments on the provided development data show that the primary system provides better performance than the best-performing system of the 2012 Albayzin evaluation (22.71% SER versus 26.34% SER in last evaluation).

Acknowledgements. This work is part of the project "Linguistic and sensorial accessibility: technologies for voiceover and audio description", funded by the Spanish Ministerio de Economía y Competitividad (FFI2012-31023). This article is supported by the Catalan Government Grant Agency Ref. 2014SGR027.

References

1. Freesound.org, `https://www.freesound.org/`
2. Tecnoparla project, `http://tecnoparla.talp.cat/`
3. The 2009 (RT-09) Rich Transcription Meeting Recognition Evaluation Plan, `http://www.itl.nist.gov/iad/mig/tests/rt/2009/docs/rt09-meeting-eval-plan-v2.pdf`
4. Anguera, X., Bonastre, J.F.: A novel speaker binary key derived from anchor models. In: INTERSPEECH. pp. 2118–2121 (2010)
5. Anguera, X., Bonastre, J.F.: Fast speaker diarization based on binary keys. In: Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. pp. 4428–4431 (May 2011)
6. Anguera, X., Movellan, E., Ferrarons, M.: Emotions recognition using binary fingerprints. In: IberSPEECH (2012)
7. Delgado, H., Fredouille, C., Serrano, J.: Towards a complete binary key system for the speaker diarization task. In: INTERSPEECH (2014)

8. Hernández-Sierra, G., Bonastre, J.F., Calvo de Lara, J.: Speaker recognition using a binary representation and specificities models. In: Alvarez, L., Mejail, M., Gomez, L., Jacobo, J. (eds.) Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Lecture Notes in Computer Science, vol. 7441, pp. 732–739. Springer Berlin Heidelberg (2012)
9. Hu, G.: 100 non-speech environmental sounds, `http://www.cse.ohio-state.edu/dwang/pnl/corpus/HuCorpus.html/`
10. Ortega, A., Castan, D., Miguel, A., Lleida, E.: The Albayzin 2014 Audio Segmentation Evaluation. In: IberSPEECH (2014)