Multiclass Digital Audio Segmentation with MFCC Features using Naive Bayes and SVM Classifiers

Leonardo O. Iheme, Şükrü Ozan
AdresGezgini Inc., Research & Development Center, Izmir, Turkey
{leonardoiheme, sukruozan}@adresgezgini.com

***Abstract*—In the field of digital audio processing, the classification of audio segments is a crucial pre-processing step towards performing more complex tasks such as automatic speech recognition or music genre classification. In our study, we investigate the use of bag of audio words, Naive Bayes and Support Vector Machines with Linear Kernel for the purpose of classifying audio segments into one of three main classes namely: silence, speech, and music. In addition, we compare the effect of using Mel Frequency Cepstral Coefficients (MFCC), their derivative and second derivative as features for both segmentation algorithms. Tests were carried out on a sample obtained from our call center database of call recordings. The results which are presented as accuracy score and Receiver Operation Characteristic (ROC) curve reveal the best use case of the chosen combination of features and segmentation algorithms.**

Keywords*—*Support Vector Machines; Navie Bayes; Bag of Audio Words; Mel Frequency Cepstral Coefficients.

I. Introduction

A technique, in which an audio stream is divided into homogeneous (similar) regions, is called audio segmentation [1]. In general, to perform digital audio content analysis, it is imperative to segment the audio stream before further processing can be carried out. Such analysis could include Automatic Speech Recognition (ASR), music genre classification, industrial machinery testing or fault detection and a host of other applications. The need for automating such audio processing tasks has risen because of rapid technological advances. Data being generated by digital audio recording devices has become overwhelming and impossible to analyze manually.

Segmentation may be rule-based or machine learning (ML) based. For simple segmentation problems such as discriminating sound from silence, rule-based segmentation algorithms produce excellent results [2]. However, in a multiclass scenario, where the classes are not easily separable, more sophisticated algorithms are required. Specifically, ML algorithms where complex models are trained with descriptive features of the samples are applied. In the field of digital audio processing, the most common features extracted are the Mel Frequency Cepstral Coefficients (MFCCs) because they sufficiently model human hearing and have performed very well in audio processing-related tasks [3].

Regarding ML for audio segmentation, the focus has been on semi-supervised learning where clustering is performed before classification. One of such schemes was employed in [4] where the authors described the Bag-of-Audio-Words (BoAW) approach for multimedia event classification. It involves using the k-means algorithm to cluster the MFCC features in order to generate a so-called codebook which is in turn fed into a classifier. The codebook can be viewed as a high-level feature space in which the most similar features are aggregated to aid classification. The set-up is flexible since it allows the use of any clustering algorithm and any classification method of one's choosing. In fact, the authors compared the performance of different support vector machines (SVM) kernels.

SVMs, in various forms, have been successfully applied to audio segmentation problems [5], [4] [6]. In [5], the authors applied a bagged SVM approach to discriminate between speech and non-speech. The zero-crossing rate, spectrum flux, short time energy and 12 MFCCs were used as features. In the study carried out in [4], MFCCs, their first and second derivatives along with the log energies were concatenated to form the feature vector. The performances of the linear kernel, radial basis function (RBF) and the histogram intersection kernel were compared.

In this work, we present the methodology and results of comparing two audio segmentation algorithms namely: bagging multiclass one-versus-rest linear SVM classifier and a BoAW naive Bayes classifier using MFCCs and concatenations of the first and second derivatives as feature vector. To the best of our knowledge, such a comparison has yet to be made. Thus, they serve as the contributions of this work. The organization of the paper is as follows: Section II provides an overview of the data, how it was collected and pre-processed; it is followed by brief descriptions of the classifications in Section III-A. The results, discussions, conclusions and feature work are presented in Sections IV, V and VI respectively.

II. Data Collection & Pre-processing

In this section, the data used in our experiments and the methodology applied are described.

A. Data-set

Our data-set is drawn from a population of call center telephone conversations generated at our local site. Each recording is sampled at 8000 Hz, single channel and stored in the compressed .gsm format. 72 files were randomly selected and manually segmented by domain experts. The manual segmentation involved annotating homogeneous regions of speech, music/tone or silence/background noise. The annotations were

978-1-7281-2868-9 / 19 / $ 31.00

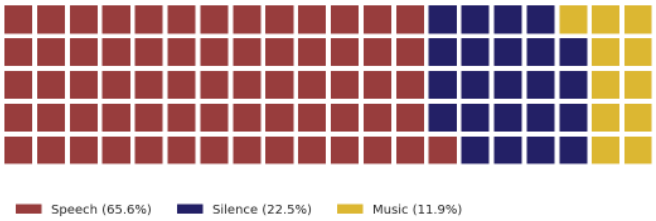

Figure 1: The unbalanced distribution of labels (speech, music/tone and silence/background noise) in the data-set

carefully carried out with the aid of version 2.3.2 of Audacity® recording and editing software.

The labelled data was stored in the following format: *Start_time Stop_time Label* as text files. Where *Start_time* refers to the corresponding time-point at which the homogeneous segment begins and *Stop_time Label* is the time-point that it ends. *Label* refers to either speech, music/tone or silence/background noise. After annotation, the labels of the data-set were distributed as shown in Figure 1.

B. Feature Extraction

1) Mel Frequency Cepstral Coefficients (MFCC): Features were extracted from the audio files in order to train classifiers. Specifically, MFCCs were extracted from each audio file. The Mel frequency cepstrum represents the short-term power spectrum of a sound. It is based on a linear cosine transform of a log power spectrum on a nonlinear Mel scale of frequency.The cepstrum can be thought of as a "spectrum of the spectrum" and provides information on how the spectrum energy changes over time. The MFCCs are the coefficients that make up the Mel cepstrum [4].

The MFCCs extracted in our experiments are in line with those described in [4]. We compare the performance of three feature vectors outlined in Table I

Feature	Description	Dimension
MFCC	12 MFCC + log energy	13
MFCC+MFCC-δ	MFCC and first derivative	26
MFCC+MFCC-δ-δ	MFCC, first derivative and second derivative	39

Table I: Extracted features, their description, and dimensions.

The feature vectors were split into 80% training and 20% development sets in order to carry out cross validation.

2) Bag of Audio Words & Naive Bayes: The BoAW technique was adapted from natural language processing (NLP) where it is used as a form of document representation. In audio processing, the procedure is similar. Since many classification techniques only take fixed vector lengths as input, the BoAW technique solves this issue by producing fixed-length histograms known as a word vectors. The procedure for computing the word vectors from MFCC vectors is as follows:

1) Code book generation using a clustering algorithm
2) Quantization of original feature vectors
3) Histogram of codewords computation

Detailed description and parameter selection criteria are further elaborated in [4] and [7].

III. Classification Algorithms

Two classification algorithms are presented in this work: Naive Bayes and SVMs. In this section, the usage of both algorithms is explained.

A. Naive Bayes

Naive Bayes is a well established classification algorithm and has been successfully applied to various problems. It is based on the Bayes theorem (Equation 1) with the assumption that the features are independent.

$$P(y \mid X) = \frac{P(X \mid y)\, P(y)}{P(X)} \tag{1}$$

In 1, y represents the labels (silence, speech or music) and X represents the features, in this case it represents the histogram obtained from BoAW. The size of X corresponds to the number of bins in the histogram.
If

$$X = (x_1, x_2, x_3, ..., x_n) \tag{2}$$

Then,

$$P(y \mid x_1, ..., x_n) = \frac{P(x_1 \mid y)P(x_2 \mid y)...P(x_n \mid y)\, P(y)}{P(x_1)P(x_2)...P(x_n)} \tag{3}$$

In a general form,

$$P(y \mid x_1, ..., x_n) \propto \prod_{i=1}^{n} P(x_i \mid y) \tag{4}$$

Finally, predictions can be made using

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i \mid y) \tag{5}$$

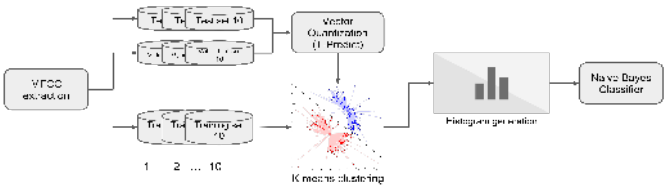

Figure 2: The bag of audio words and naive Bayes framework

The setup for BoAW and naive Bayes is summarized in Figure 2

B. Support Vector Machines

SVM constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier [8]. Details of the formulation of the SVM algorithm have been left out

because of page limit restrictions. However, for further reading about SVM, refer to [8].

SVMs have been successfully applied to segmentation problems from other fields of study including digital image processing. In this work, 10 linear kernel SVMs are applied in a bagging classifier [9] using the one-versus-rest technique to tackle the multiclass problem.

IV. Experimental Set-up and Results

An 80-20% train-test split and 10-fold cross validation was performed to tune the model and test its performance. However, due to the unbalanced nature of the data as depicted in Figure 1, the experiments were carried out for the balanced and unbalanced data sets separately. To balance the data set, the sizes of the *speech* and *silence* samples were reduced to match the *music* samples.

In the feature matrix, each sample corresponds to a MFCC frame which translates to 25 milliseconds of audio. Realistically, such a small duration is not sufficient to characterize speech, silence or music. Furthermore, the ground truth data was labeled as a collection of frames and not on the basis of a single frame. Therefore, we introduce a parameter, $T_{predict}$ which is the optimum number of frames that are required to predict an instance of any of the classes. In the experiments, $T_{predict}$ was also optimized. Lastly, while building the BoAW, the code book size K was also varied to obtain an optimal value.

A. Evaluation

The models were evaluated by means of the true and false positive rates. Receiver operating characteristics (ROC) were plotted and the area under the curves (AUC) were calculated. In addition, accuracy scores were computed for each model. All experiments were carried out for balanced and unbalanced data.

B. Results

We present the results of evaluating the BoAW + naive Bayes and SVM algorithms on the same training and testing data. The results are presented as accuracy scores and ROC of each model.

1) Support Vector Machines: The mean accuracy and AUC of the three classes (speech, music, and silence) for the unbalanced and balanced data-sets are presented. The experiments were carried out for the different feature sets mentioned in sub-section II-B and the results presented were obtained from the optimal hyper-parameter values.

In Figure 3, the mean accuracy obtained for the balanced/unbalanced data set and the three feature sets is shown. Table II shows the macro average AUC of the three classes and the six different scenarios of the experimental set up. The highest accuracy is obtained when we train the model with the unbalanced data set and the MFCC, δ, and $\delta - \delta$ features.

A more detailed AUC performance plot is depicted in Figures 4 and 5 where ROC curves are plotted for the individual classes as well the micro- and macro-averages.

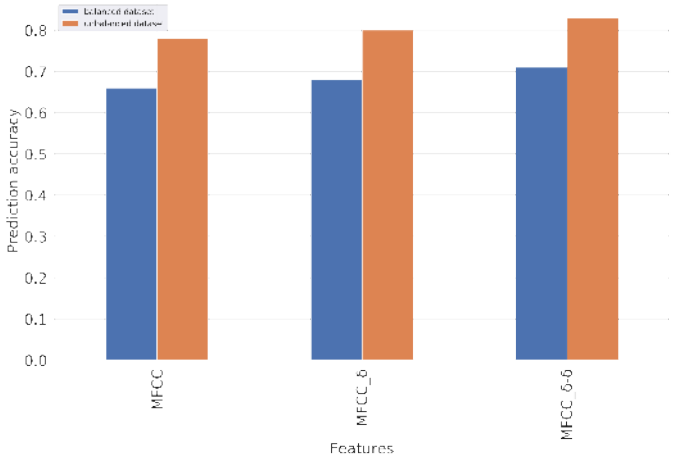

Figure 3: Bar graph showing the accuracy of the SVM classifier for the different scenarios of features and balanced/unbalanced data-sets

Measure	Features	Balanced	Unbalanced
AUC	MFCC	0.77	0.81
	δ	0.79	0.83
	$\delta - \delta$	0.82	**0.86**

Table II: Performance of the SVM classifier based on the three feature vectors and the balanced/unbalanced data set

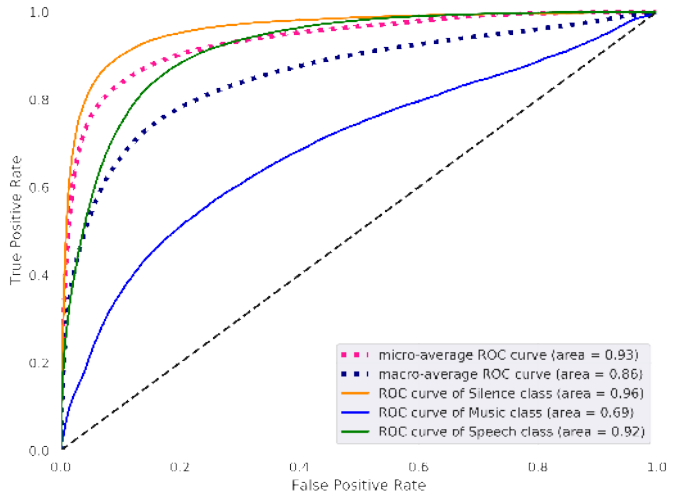

Figure 4: Receiver Operating Characteristics (ROC) curves for the one-versus-rest ensemble bagging linear kernel SVM classifier with the unbalanced data-set

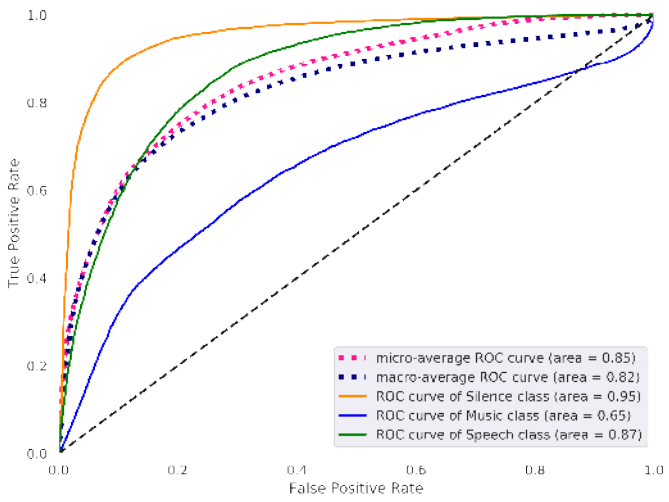

Figure 5: Receiver Operating Characteristics (ROC) curves for the one-versus-rest ensemble bagging linear kernel SVM classifier with the balanced data-set

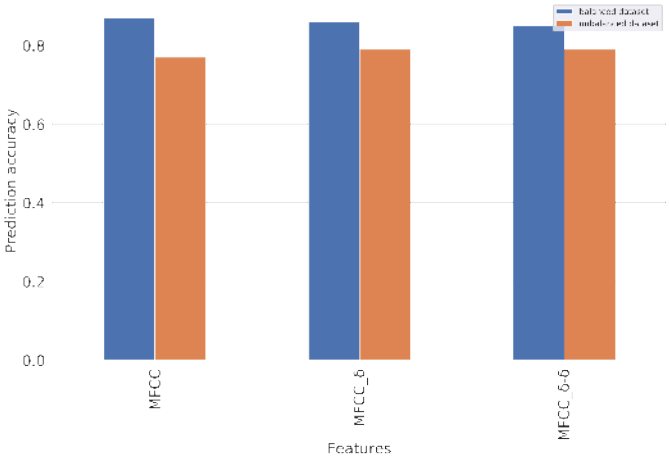

Figure 6: Bar graph showing the accuracy of the bag of audio words – naive Bayes classifier for the different scenarios of features and balanced/unbalanced data sets

2) Bag of Audio Words – Naive Bayes: After optimizing the parameters K and $T_{predict}$ via cross validation, the results are presented in terms of accuracy and AUC. Figure 6 shows how the models perform in terms of prediction accuracy on the balanced and unbalanced data sets. The average AUC remained unchanged at 0.95 for this set of experiments.

V. Discussion

The results obtained express the predictive capability of the models which were built. At best the SVM classifier attained an accuracy of 0.83 with the unbalanced data set and the MFCC $\delta - \delta$ features - significantly higher than accuracies obtained for the balanced data set (0.71). A proportional relationship is observed between the feature complexity and the accuracy of the model for both data sets, i.e. the more complex the features are, the higher the accuracy of the prediction. This can be attributed to the fact that higher order MFCCs capture more descriptive features and SVM is known to perform relatively well even for unbalanced data. The somewhat brute-force technique used to balance the data greatly reduced the number of training samples available. With much fewer training examples, the model bias was increased leading to under fitting.

Regarding the BoAW-naive Bayes classifier, the optimum accuracy (0.87) was achieved when the data set was balanced and basic MFCC features were used for classification. Unlike the SVM algorithm, the accuracy seemed to deteriorate as the complexity of the features increased. The range of accuracies obtained across the three feature sets is small: 0.02, for both the balanced and unbalanced data sets while the AUC (0.95) remained the same for all the experiments. This raises the question: does the order of the MFCC features affect the performance of the BoAW-naive Bayes model? The answer lies in performing in-depth statistical analysis which is beyond the scope of this article. The superior performance on the balanced data set is due to how naive Bayes handles the zero-frequency problem associated with BoAW as explained in [10]. Although, balancing the data improved the performance, it came at the cost of a loss of information from the majority classes.

VI. Conclusion & Feature Work

This study presented the comparison of SVM and BoAW-naive Bayes segmentation algorithms applied to call center audio data. The performance of each algorithm was evaluated by the accuracy score and the area under the receiver operating characteristics curve. For the SVM algorithm, the best accuracy score, which was 0.83, was achieved when MFCC features were concatenated with the first and second derivatives. This accuracy was achieved by training with the unbalanced data set which shows that SVM is relatively robust to data unbalance. The reverse was the case with the BoAW-naive Bayes algorithm were balancing the data and using fewer features yielded a better performance.

In future studies, the results of the two algorithms may be improved by:

- Providing a balanced data set to be used for training or using a more sophisticated technique to balance the data
- Testing different SVM kernels such as the RBF or polynomial kernel and optimizing their hyper-parameters

While SVM has yielded satisfactory results on its own, it will be worth examining the outcome of replacing the naive Bayes classifier with a SVM instead. Such a configuration was applied in [4] and the results were presented in terms of Detection Error Trade-off (DET) curves.

Acknowledgements

This study is supported by the Scientific and Technological Research Council of Turkey under the grant TEYDEB 1507, project number 7170694.

REFERENCES

[1] A. Lerch, *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*, 1st ed. Wiley-IEEE Press, 2012.

[2] W. Qing Ong, A. Wee Chiat Tan, V. Vijayakumar Vengadasalam, C. Heng Tan, and T. Hai Ooi, "Real-time robust voice activity detection using the upper envelope weighted entropy measure and the dual-rate adaptive nonlinear filter," *Entropy*, vol. 19, p. 487, 10 2017.

[3] M. Xu, L.-Y. Duan, J. Cai, L.-T. Chia, C. Xu, and Q. Tian, "Hmm-based audio keyword generation," in *Advances in Multimedia Information Processing - PCM 2004*, K. Aizawa, Y. Nakamura, and S. Satoh, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 566–574.

[4] S. Pancoast and M. Akbacak, "Bag-of-audio-words approach for multimedia event classification," in *INTERSPEECH*, 2012.

[5] S. Zahid, F. Hussain, M. Rashid, M. H. Yousaf, and H. A. Habib, "Optimized audio classification and segmentation algorithm by using ensemble methods," *Mathematical Problems in Engineering*, vol. 2015, 05 2015.

[6] S. E. Krüger, M. Schafföner, M. Katz, E. Andelic, and A. Wendemuth, "Speech recognition with support vector machines in a hybrid system," in *in Proc. EuroSpeech, 2005*, 2005, pp. 993–996.

[7] M. Schmitt, F. Ringeval, and B. W. Schuller, "At the border of acoustics and linguistics: Bag-of-audio-words for the recognition of emotions in speech," in *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, 2016, pp. 495–499. [Online]. Available: https://doi.org/10.21437/Interspeech.2016-1124

[8] Wikibooks, "Support vector machines — wikibooks, the free textbook project," 2018, [Online; accessed 1-July-2019]. [Online]. Available: \url{https://en.wikibooks.org/w/index.php?title=Support_Vector_Machines&oldid=3454247}

[9] L. Breiman, "Pasting small votes for classification in large databases and on-line," *Machine Learning*, vol. 36, no. 1, pp. 85–103, Jul 1999. [Online]. Available: https://doi.org/10.1023/A:1007563306331

[10] E. Frank and R. R. Bouckaert, "Naive bayes for text classification with unbalanced classes," in *Knowledge Discovery in Databases: PKDD 2006*, J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 503–510.