INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Partial AUC Optimisation using Recurrent Neural Networks for Music Detection with Limited Training Data

Pablo Gimeno, Victoria Mingote, Alfonso Ortega, Antonio Miguel, Eduardo Lleida

ViVoLab, Aragón Institute for Engineering Research (I3A), University of Zaragoza
{pablogj, vmingote, ortega, amiguel, lleida}@unizar.es

Abstract

State-of-the-art music detection systems, whose aim is to distinguish whether or not music is present in an audio signal, rely mainly on deep learning approaches. However, these kind of solutions are strongly dependent on the amount of data they were trained on. In this paper, we introduce the area under the ROC curve (AUC) and partial AUC (pAUC) optimisation techniques, recently developed for neural networks, into the music detection task, seeking to overcome the issues derived from data limitation. Using recurrent neural networks as the main element in the system and with a limited training set of around 20 hours of audio, we explore different approximations to threshold-independent training objectives. Furthermore, we propose a novel training objective based on the decomposition of the area under the ROC curve as the sum of two partial areas under the ROC curve. Experimental results show that partial AUC optimisation can improve the performance of music detection systems significantly compared to traditional training criteria such as cross entropy.

Index Terms: music detection, recurrent neural networks, partial AUC optimisation, limited training data

1. Introduction

In the last few years, we have observed how audiovisual repositories are becoming larger and larger, making the manual annotation and tagging unfeasible in some cases. That is the reason why automatic systems that can extract information in an accurate way are becoming significantly relevant. In this paper, we focus on the music detection task, whose aim is to determine whether or not music is present in an audio excerpt. Music detection is especially relevant in the broadcast domain, where music is usually mixed with speech and other type of non-music sounds. Besides from the automatic indexing and retrieval of information based on the audio content, music detection plays an important role in broadcast emissions in the context of monitoring for copyright management [1] [2].

Traditionally, statistical approaches were applied to the music detection task. In [3], support vector machines were used to separate speech and music in radiophonic streams. The same objective of discriminating speech and music was solved in [4] using multi-stage decision trees. The factor analysis technique was applied in [5] to detect simultaneously speech, music and noise with relevant results for broadcast domain data.

Currently, state of the art music detection systems are mainly based on deep learning approaches. In [6], authors explore the use of convolutional and recurrent neural networks for both music and speech detection applied to a large audio dataset. A Mel-scale kernel is proposed in [7] to be used in a convolutional neural network for broadcast music detection. A semi-supervised approach is presented in [8], where convolutional neural networks are used to classify speech and music. Our previous experience in audio segmentation [9][10] showed the feasibility of recurrent neural networks for multiclass segmentation concerning speech, music and noise. The set of technological evaluations MIREX [11] proposed in 2018 a music detection task under different conditions. The best performing system [12] was based on a convolutional neural network with Mel spectrograms as input.

One of the main disadvantages of deep learning systems is that their performance is strongly dependent on the data they were trained on. In this paper, we aim to overcome this issue when using a limited amount of training data by introducing the recently presented area under the ROC curve optimisation techniques for neural networks. Recent studies that are discussed in the following sections have proved that they outperform traditional training objectives such as cross entropy in other detection tasks.

The remainder of the paper is organised as follows: Section 2 introduces the AUC and pAUC optimisation framework that we propose for the music detection task. Section 3 presents the experimental setup, describing the neural network architecture, the datasets considered and the metrics used in the evaluation. In Section 4, we describe the results obtained for our music detection system. Finally, a summary and the conclusions are presented in Section 5.

2. AUC and pAUC optimisation framework

2.1. Problem formulation

Suppose a dataset $\Pi = \{\mathbf{X}, \mathbf{Y}\}$ where $\mathbf{X} = \{x_1, \dots, x_N\}$ is the set of acoustic features with N different examples, and $\mathbf{Y} = \{y_1, \dots, y_N\}$ the music labels defining each of the elements in $\mathbf{X}$ as music or non-music examples (1 or 0 respectively). The neural network can be expressed then as a function $f_\theta : \mathbb{R}^D \rightarrow \mathbb{R}$ depending on a set of parameters θ and mapping the input space of dimension D to a real number representing the music score. The parameters θ are estimated iteratively through the backpropagation algorithm seeking to minimise (or maximise) a metric given by a loss function $L(f_\theta(x_i), y_i)$ that measures the difference (or similarity) between the neural network output and the labels.

2.2. Area under the ROC curve optimisation

The receiver operating characteristic (ROC) curve is a well known method to represent the performance of a detection system. This curve plots false positive rates (FPR) versus true positive rates (TPR) for all the possible detection thresholds. Furthermore, the area under the ROC curve (AUC) measures the area underneath the entire ROC curve, providing an aggregate performance measure which is independent of the detection threshold. Depending on the desired behaviour, the detection threshold may be chosen differently. That is why it would

http://dx.doi.org/10.21437/Interspeech.2020-1108

be desirable to optimise the system for all the possible decision thresholds, taking into account the trade-off between false positives and false negatives.

Several papers already proposed to directly optimise the AUC for different applications with promising results [13][14]. Focusing on those using neural networks, an AUC optimisation framework is adopted in [15] for the text-dependent speaker verification task. In [16], a deep learning based speech activity detection system is trained with an AUC optimisation criterion. To the best of our knowledge, this is the first approach to AUC optimisation in the music detection task.

To compute the AUC metric two new subsets need to be defined: $\mathbf{S}^+ = \{f_\theta(x_i)\ \forall x_i \in \mathbf{X}\,|\,y_i = 1\}$, which is the neural network scores for the positives examples in $\mathbf{X}$, and $\mathbf{S}^- = \{f_\theta(x_i) \forall x_i \in \mathbf{X}\,|\,y_i = 0\}$ that represents the neural network scores for the negatives examples in $\mathbf{X}$. Cardinalities of those sets are N^+ and N^- respectively. Then, the AUC loss can be defined as

$$L_{AUC} = \frac{1}{N^+N^-}\sum_{i=1}^{N^+}\sum_{j=1}^{N^-}\mathbb{1}\Big(s_i^+ > s_j^-\Big)\,, \tag{1}$$

where $\mathbb{1}(\cdot)$ is equal to '1' whenever $s_i^+ > s_j^-$ and '0' otherwise. This expression can be rewritten using the unit step function as

$$L_{AUC} = \frac{1}{N^+N^-}\sum_{i=1}^{N^+}\sum_{j=1}^{N^-}u\Big(s_i^+ - s_j^-\Big)\,. \tag{2}$$

In order to enable the backpropagation of the gradients, an approximation must be done to obtain a differentiable function. In our implementation, we adopt the expression proposed in [15], that modifies the step function for a sigmoid function according to

$$L_{aAUC} = \frac{1}{N^+N^-}\sum_{i=1}^{N^+}\sum_{j=1}^{N^-}\sigma\Big(\delta(s_i^+ - s_j^-)\Big)\,, \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid function and δ is an hyperparameter that controls the slope of the sigmoid.

2.3. Partial AUC optimisation

The usefulness of AUC can be limited in some cases due to its threshold invariance. If there are wide disparities in the cost of false positives versus false negatives, it may be critical to minimise one type of error. Furthermore, optimising the whole ROC curve can be costly and, in some specific applications, needless as the system is going to operate only in a certain area of the ROC curve. The partial AUC (pAUC) overcomes these issues evaluating the performance for a region of interest of FPR values. It is formally defined as the area under the ROC curve between two FPR values α and β. This metric is presented schematically in Figure 1, with the grey area representing the pAUC.

pAUC optimisation was firstly proposed in [17] for the speaker verification task outperforming the current state-of-the-art on the NIST 2016 SRE data. The hard negative mining approach presented in [15] could also be interpreted as a partial AUC solution that chooses the hardest examples for training. A more recent study [18] has compared both AUC and pAUC training objectives to obtain speaker embeddings for text-independent speaker verification. This work shows that pAUC training achieves better results than AUC training in most cases.

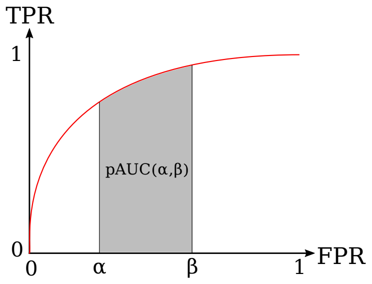

Figure 1: *Schematic representation of ROC curve and partial AUC given α and β.*

Given the already defined $\mathbf{S}^-$ set, a new set $\mathbf{S}^-_{\alpha\beta}$ must be obtained constraining $\mathbf{S}^-$ to the range where the false positive rate lies in the interval $[\alpha, \beta]$. This is done through the following three main steps:

1. The interval $[\alpha, \beta]$ must be replaced for its integer equivalent, this is $[\frac{n_\alpha^-}{N^-}, \frac{n_\beta^-}{N^-}]$. with n_α^- and n_β^- computed according to
$$n_\alpha^- = \lceil \alpha N^- \rceil + 1\,, \qquad n_\beta^- = \lfloor \beta N^- \rfloor\,. \tag{4, 5}$$
2. Sort $\mathbf{S}^-$ in descending order
3. $\mathbf{S}^-_{\alpha\beta}$ is selected as the set of samples ranked from the top n_α^-th to the n_β^-th position of the sorted $\mathbf{S}^-$ set. This results in a set of length $N^-_{\alpha\beta} = n_\beta^- - n_\alpha^- + 1$

Now, the partial AUC can be computed in a similar way to the expression proposed in Eq. 3, but substituting the set of negative examples $\mathbf{S}^-$ for the new set of constrained negative examples $\mathbf{S}^-_{\alpha\beta}$. This partial AUC is expressed as

$$L_{apAUC}(\alpha,\beta) = \frac{1}{N^+N^-_{\alpha\beta}}\sum_{i=1}^{N^+}\sum_{j=1}^{N^-_{\alpha\beta}}\sigma\Big(\delta(s_i^+ - s^-_{\alpha\beta_j})\Big)\,. \tag{6}$$

2.4. AUC optimisation as sum of two partial AUCs

When computing the pAUC loss, a fraction of the training examples is discarded. This is done after sorting the $\mathbf{S}^-$ set, where only a subset of the sorted set is used in training. If we suppose $\alpha = 0$ as done in [18], a fraction $1 - \beta$ of the examples are consistently dropped in the training process. This fact could be seen as a way of speeding up training because it is reducing the number of operations per iteration. However, as we are dealing with a limited training data scenario, we believe that it would be interesting to incorporate those discarded examples in training somehow.

Our idea to incorporate those examples in training is presented schematically in Figure 2. The proposed training objective decomposes the entire AUC as the sum of two partial AUC, assuming the first one is using $\alpha = 0$ and the second one is using $\beta = 1$. The parameter γ is introduced as the FPR point that separates the area in two subareas. Then, our new training objective can be computed as

$$L_{aAUCsum}(\gamma,\lambda) = L_{apAUC}(0,\gamma) + \lambda L_{apAUC}(\gamma,1)\,. \tag{7}$$

A scalar hyperparameter λ is used for balancing both parts, seeking to give more or less importance to the second pAUC in training.

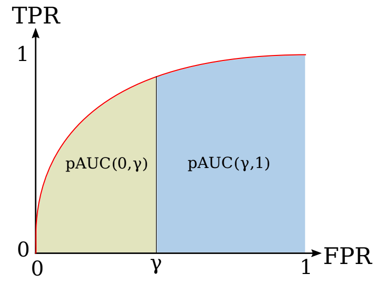

Figure 2: *Schematic representation of ROC curve and our proposed computation of AUC as the sum of two partial AUC defined by the parameter γ.*

3. Experimental setup

3.1. Neural network

In this work, the neural architecture is based on recurrent neural networks (RNN). We stack 2 bidirectional gated recurrent units (GRU) [19] with 128 neurons each, followed by a linear layer that performs the final classification. Adam optimiser is used with a learning rate that decays exponentially from 10^{-3} to 10^{-4} during the 20 epochs that data is presented to the neural network. The hyperparameter δ, introduced in subsection 2.2 and used in the AUC based optimisation experiments, has a fixed value of 10 seeking to obtain a shape close to the unit step function, in a similar way as it is done in [15]. Training and evaluation is done using limited length sequences (3 seconds, 300 frames) in order to limit the delay of dependencies to take into account by the RNN. However, an output label is emitted for every frame processed at the input.

Concerning feature extraction, inspired by our previous experience in audio segmentation tasks [9][10], we combine a traditional set of perceptual features with some musical theory motivated features. First, 128 log Mel filter bank energies are extracted between 64 Hz and 8kHz. Additionally, they are combined with chroma features [20], a projection of the entire spectrum into 12 bins representing the 12 distinct semitones of the chromatic musical scale. Chroma features are extracted using the openSMILE toolkit [21]. All features are computed every 10 ms using a 25 ms Hamming window. Before being concatenated, log Mel energies and chroma features are first normalised to be in the range between 0 and 1.

This setup is fixed for all our experiments as our main goal is not to evaluate this neural architecture compared to other proposals, but to evaluate whether AUC and pAUC optimisation can improve the performance of our music detection system over traditional training objectives.

3.2. Data description

In order to perform our experiments we consider two different datasets, one for training and another for testing. Both of them correspond to the broadcast domain, coming from different television emissions. In the following lines we describe the two datasets:

- **Train**: we use the recently released OpenBMAT[1] dataset [22] to train our music detection system. It contains 27 hours of television broadcast audio from different countries labelled by 3 annotators for the music detection task. High inter-annotator agreements ratios were obtained so, for training we only considered the labels from the first annotator. These labels contain a 51% of music and a 49% of non-music. Furthermore, a 10% of the data is reserved for training validation, so the total amount of data used for training is around 24 hours. No prior data picking is performed in any case; minibatches are sampled randomly for the training dataset.
- **Test**: as test data we use the dataset[2] originally presented in the paper "Automatic Music Detection in Television Productions" [23]. It consists of around 9 hours of television recordings from the Austrian national broadcasting corporation that were manually labelled as music or non-music. There is a great variety of genres, ranging from soap operas to documentaries or talk shows. The data distribution is close to be equally distributed, with 42% of the time being music and 58% of the time being non-music.

All the audios have been downsampled to 16 kHz and mixed down to a single channel. Both datasets can be downloaded for research purposes upon request to their respective authors.

3.3. Evaluation metrics

The music detection task can be interpreted as a binary classification task, so traditional metrics for binary tasks are applicable. In this paper we evaluate the results of our system using the AUC metric, which measures the area underneath the entire ROC curve, and the equal error rate (EER), the error rate at which the false negative rate and false positve rate is equal. We also present the complete ROC curve with true positive rates versus false positive rates for the best systems presented in the paper.

Additionally, we report in our results recall and F_1 measure for a system using a threshold so that precision = 0.90, with precision, recall and F_1 measure computed according to

$$\mathrm{Prec} = \frac{tp}{tp + fp}\,, \qquad \mathrm{Rec} = \frac{tp}{tp + fn}\,, \qquad (8, 9)$$

$$\mathrm{F}_1 = 2\frac{\mathrm{Prec}\cdot\mathrm{Rec}}{\mathrm{Prec} + \mathrm{Rec}}\,, \qquad (10)$$

where tp is the number of true positive predictions, fp is the number of false positive predictions and fn is the number of false negative predictions. All metrics are computed at frame level and without collar on the full test data presented in the previous subsection.

4. Results

As the starting point of our experimentation, our aim was to obtain a baseline system so that our further results could be compared. With the experimental framework described in Section 3, we trained a neural network using the well-known cross entropy loss. This system serves as a point of comparison for our AUC and pAUC optimisation experiments. In Table 1, we compare this baseline with a system trained using the aAUC training criterion. It can be observed that, by shifting from traditional loss functions, such as cross entropy, to the aAUC criterion, a significant 4.30% relative improvement can be observed in terms of AUC and a 19.11% relative improvement in terms of EER.

Once it has been proved that AUC optimisation can improve music detection performance in a limited data scenario

[1] https://zenodo.org/record/3381249#.XfNfDNmCFuU

[2] http://www.seyerlehner.info/index.php?p=1_3_Download

Table 1: *AUC, EER, recall and F_1 measure (both computed for a system with precision fixed at 0.90) on test data for the music detection system trained using the aAUC training criterion compared to a cross entropy based training.*

Training criterion	**AUC(%)**	**EER(%)**	**Rec**	**F_1**
Cross entropy	87.02	21.40	0.45	0.60
aAUC	**90.76**	**17.31**	**0.65**	**0.75**

Table 2: *AUC, EER, recall and F_1 measure (both computed for a system with precision fixed at 0.90) on test data for the music detection system trained using the apAUC training objective and different values of α and β*

apAUC optimisation	**AUC(%)**	**EER(%)**	**Rec**	**F_1**
$\alpha = 0$ $\beta = 0.25$	87.21	20.85	0.54	0.68
$\alpha = 0$ $\beta = 0.5$	90.34	17.53	0.64	0.75
$\boldsymbol{\alpha = 0}$ $\boldsymbol{\beta = 0.75}$	**91.88**	**16.03**	**0.71**	**0.79**
$\alpha = 0.25$ $\beta = 0.5$	86.95	21.18	0.40	0.55
$\alpha = 0.25$ $\beta = 0.75$	88.79	19.56	0.55	0.64
$\alpha = 0.25$ $\beta = 1$	86.83	21.69	0.42	0.57

with this first experiment, in the next set of experiments we explore a more generalised training criterion, this is the apAUC explained previously. It can be easily observed that the aAUC criterion is a specific case of the apAUC criterion that uses parameters $\alpha = 0$ and $\beta = 1$. Results using the pAUC training criterion are presented in Table 2. It can be observed that two different parameter setups were assessed. The first set of parameters uses $\alpha = 0$, in a similar way as done in [18]. This configuration is equivalent to discard the fraction $1 - \beta$ of non-target examples with the lower scores, therefore the ones that are easier to classify. The best system performance is achieved for $\beta = 0.75$, with evaluation metrics even better than the ones obtained for the aAUC training objective. This results in a relative improvement compared to the baseline system of 5.60% in terms of AUC and 25.10% in terms of EER. The setup using $\alpha = 0.25$ is presented for comparison purposes. In this case the non-target examples with the higher scores (harder to classify) are discarded for training. It can be clearly observed that this setup underperforms the one with $\alpha = 0$, with AUC and EER values which are close to the baseline system.

Our final experiment explored the proposed aAUCsum training criterion that separates the AUC optimisation in two partial AUCs. The results obtained are presented in Table 3. Gamma value was chosen to be 0.75 because it obtained the best performance in the pAUC optimisation experiments. The parameter setup slightly outperforms the apAUC training objective, however, experimental results suggest that incorporating the discarded examples in training does not lead to a consistent performance increment. This effect may be motivated by the fact that the hardest examples could be sufficient for the neural network to learn an effective classification mapping. It is also remarkable that, our proposed aAUCsum criterion achieves state of the art performance, with an AUC and EER better than the aAUC training criterion in all cases. This results matches with previous studies that suggest that pAUC optimisation techniques outperform AUC techniques.

Finally, Figure 3 presents the ROC curves for the different systems trained in this paper using the best parameter configuration obtained. Again, it can be observed that the baseline system trained using cross entropy criterion is significantly below in performance compared to the other systems presented in this paper. Furthermore, a similar ROC curve is obtained for the apAUC and aAUCsum training objectives, both with a performance better than the one obtained for the aAUC training.

Table 3: *AUC, EER, recall and F_1 measure (both computed for a system with precision fixed at 0.90) on test data for the music detection system trained using the aAUCsum training objective, $\gamma = 0.75$ and different values of λ.*

aAUCsum optimisation	**AUC(%)**	**EER(%)**	**Rec**	**F_1**
$\gamma = 0.75$ $\lambda = 1$	91.78	16.78	0.70	0.79
$\gamma = 0.75$ $\lambda = 0.1$	91.31	16.59	0.68	0.77
$\gamma = 0.75$ $\lambda = 0.01$	91.45	16.20	0.69	0.78
$\boldsymbol{\gamma = 0.75}$ $\boldsymbol{\lambda = 0.001}$	**92.02**	**15.78**	**0.72**	**0.80**
$\gamma = 0.75$ $\lambda = 0.0001$	91.31	16.78	0.70	0.79

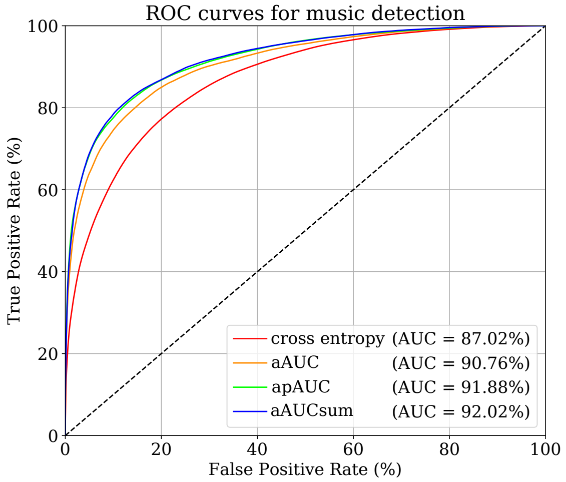

Figure 3: *ROC curve on the test data for the different training objectives presented in the paper using the best parameter configuration obtained (pAUC: $\alpha = 0$ and $\beta = 0.75$, AUCsum: $\gamma = 0.75$ and $\lambda = 0.001$)*

5. Conclusions

In this paper, we have introduced the AUC and pAUC optimisation techniques into the music detection task, aiming to overcome the issues derived from data limitations. We present several approximations to threshold-independent loss functions, using a system based on recurrent neural networks, and with a limited training data set consisting only of around 20 hours of audio.

Experimental results suggest that partial AUC training outperforms traditional training objectives such as cross entropy. In particular, we report a relative improvement close to 5.75% in terms of AUC, and 26.26% in terms of EER when using our proposed aAUCsum loss function. These results match with the previously published studies that state that pAUC optimisation techniques report better performance than AUC based optimisation.

6. Acknowledgements

This work has been supported by the Spanish Ministry of Economy and Competitiveness and the European Social Fund through the project TIN2017-85854-C4-1-R, Government of Aragón (Reference Group T36_20R) and co-financed with Feder 2014-2020 "Building Europe from Aragón". We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research.

7. References

[1] T. Izumitani, R. Mukai, and K. Kashino, “A background music detection method based on robust feature extraction,” in *Proc. IEEE ICASSP*, 2008, pp. 13–16.

[2] Y. Zhu, Q. Sun, and S. Rahardja, “Detecting musical sounds in broadcast audio based on pitch tuning analysis,” in *Proc. IEEE International Conference on Multimedia and Expo*, 2006, pp. 13–16.

[3] G. Richard, M. Ramona, and S. Essid, “Combined supervised and unsupervised approaches for automatic segmentation of radiophonic audio streams,” in *Proc. IEEE ICASSP*, 2007, pp. II–461.

[4] Y. Lavner and D. Ruinskiy, “A decision-tree-based algorithm for speech/music classification and segmentation,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2009, pp. 1–14, 2009.

[5] D. Castán, A. Ortega, A. Miguel, and E. Lleida, “Audio segmentation-by-classification approach based on factor analysis in broadcast news domain,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2014, p. 34, 2014.

[6] D. de Benito-Gorron, A. Lozano-Diez, D. T. Toledano, and J. Gonzalez-Rodriguez, “Exploring convolutional, recurrent, and hybrid deep neural networks for speech and music detection in a large audio dataset,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2019, p. 9, 2019.

[7] B.-Y. Jang, W.-H. Heo, J.-H. Kim, and O.-W. Kwon, “Music detection from broadcast contents using convolutional neural networks with a mel-scale kernel,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2019, p. 11, 2019.

[8] D. Doukhan and J. Carrive, “Investigating the use of semi-supervised convolutional neural network models for speech/music classification and segmentation,” in *The Ninth International Conferences on Advances in Multimedia (MMEDIA 2017):*, 2017.

[9] P. Gimeno, I. Viñals, A. Ortega, A. Miguel, and E. Lleida, “Multiclass audio segmentation based on recurrent neural networks for broadcast domain data,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2020, no. 1, pp. 1–19, 2020.

[10] ——, “A recurrent neural network approach to audio segmentation for broadcast domain data,” *Proc. IberSPEECH 2018*, pp. 87–91, 2018.

[11] J. S. Downie, A. F. Ehmann, M. Bay, and M. C. Jones, “The music information retrieval evaluation exchange: Some observations and insights,” in *Advances in music information retrieval*. Springer, 2010, pp. 93–115.

[12] B. Meléndez-Catalán, E. Molina, and E. Gomez, “Music and/or speech detection MIREX 2018 submission,” in *Music Information Retrieval Evaluation eX-change (MIREX)*, 2018.

[13] L. P. Garcia-Perera, J. A. Nolazco-Flores, B. Raj, and R. Stern, “Optimization of the DET curve in speaker verification,” in *2012 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2012, pp. 318–323.

[14] K.-A. Toh, J. Kim, and S. Lee, “Maximizing area under ROC curve for biometric scores fusion,” *Pattern Recognition*, vol. 41, no. 11, pp. 3373–3392, 2008.

[15] V. Mingote, A. Miguel, A. Ortega, and E. Lleida, “Optimization of the area under the ROC curve using neural network supervectors for text-dependent speaker verification,” *Computer Speech & Language*, p. 101078, 2020.

[16] Z.-C. Fan, Z. Bai, X.-L. Zhang, S. Rahardja, and J. Chen, “AUC optimization for deep learning based voice activity detection,” in *Proc. IEEE ICASSP*, 2019, pp. 6760–6764.

[17] Z. Bai, X.-L. Zhang, and J. Chen, “Partial AUC metric learning based speaker verification back-end,” *arXiv preprint arXiv:1902.00889*, 2019.

[18] ——, “Partial AUC optimization based deep speaker embeddings with class-center learning for text-independent speaker verification,” *arXiv preprint arXiv:1911.08077*, 2019.

[19] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.

[20] M. A. Bartsch and G. H. Wakefield, “To catch a chorus: Using chroma-based representations for audio thumbnailing,” in *Proc. IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*, 2001, pp. 15–18.

[21] F. Eyben, M. Wöllmer, and B. Schuller, “Opensmile: the munich versatile and fast open-source audio feature extractor,” in *Proc. 18th ACM international conference on Multimedia*, 2010, pp. 1459–1462.

[22] B. Meléndez-Catalán, E. Molina, and E. Gómez, “Open broadcast media audio from TV: A dataset of TV broadcast audio with relative music loudness annotations,” *Transactions of the International Society for Music Information Retrieval*, vol. 2, no. 1, 2019.

[23] K. Seyerlehner, T. Pohle, M. Schedl, and G. Widmer, “Automatic music detection in television productions,” in *Proc. 10th International Conference on Digital Audio Effects*, 2007.