AMBIGUITY MODELLING WITH LABEL DISTRIBUTION LEARNING FOR MUSIC CLASSIFICATION

Morgan Buisson, Pablo Alonso-Jiménez, Dmitry Bogdanov

Music Technology Group, Universitat Pompeu Fabra

ABSTRACT

An important amount of work has been devoted to the task of music classification. Despite promising results achieved by convolutional neural networks, there still exists a gap left to be filled for such models to perform well in real-world applications. In this work, we address the issue of ambiguity that can arise in many classification problems. We propose a method based on adaptive label smoothing that aims at implicitly modelling perceptual vagueness among classes to improve both training and testing performances. We assess our method using two state-of-the-art CNN architectures for audio classification on a variety of music mood and genre classification tasks. We show that the proposed strategy brings consistent improvements over the traditional approach, significantly improves generalization to external audio collections and emphasizes how crucial information carried by labels can be in an ambiguous music classification context.

Index Terms— Music classification, Deep learning, Label distribution learning, Generalization

1. INTRODUCTION

Automatic music classification has been extensively addressed and remains an active area of research. Although recent improvements have been possible through the use of convolutional neural networks (CNN) architectures appropriate for such tasks [1–3], abstract notions such as genre or mood depend on high-level characteristics that most existing models fail at capturing explicitly. Many classification problems can be subject to ambiguity among classes and disagreements among raters when asked to provide ground truth. As most of the current classifiers follow supervised learning paradigms, their performance is directly dependent on the quality of the labels they are provided with during training.

In this work we propose a method that aims at carefully transforming labels distribution to limit overfitting in small datasets scenarios and ultimately improve generalization to out-of-sample data. We reformulate the task of classification as a label distribution learning problem. To obtain a probability distribution over all classes for each track, we learn a linear projection of the data that better approximates the ambiguity of each song. This paper is organized as follows: we first define the notions of ambiguity and subjectivity and introduce the label distribution learning technique; we then describe in more detail the method we propose and present the different experiments performed to assess its efficiency; finally, we discuss the overall feasibility of our approach and give further axes of improvements.

2. BACKGROUND

The notion of ambiguity can be formally described as an uncertainty among ground-truth labels. In some cases, label ambiguity is inherent to the task and obtaining accurate labels turns out to be a complex and exhausting process. Similarly, label subjectivity refers to the implication of raters' feelings, perception or experiences into the annotation process. When building a dataset, if neither of these elements are taken into account, the resulting label attribution of all training examples might fail at representing the inherent nature of the classification task. To alleviate this issue, previous studies in domains such as age recognition [4], head-pose estimation [5] or semantic segmentation [6] have benefited from an explicit consideration of ambiguity through label distribution learning. The same conclusion could potentially be drawn in the context of music classification, where concepts such as mood or genre heavily depend on perceptual, cultural and representation aspects [7–10].

Formally introduced by Gao et al. in [11], the label distribution learning (LDL) paradigm aims at converting original labels to discrete label distributions. Let $\mathcal{X} = \mathbb{R}^n$ denote the input space and $\mathcal{Y} = \{y_i\}_{i=1}^K$ denote the complete set of labels. Given a training set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, the goal of LDL is to learn a conditional probability mass function $p(y \mid \mathbf{x})$ from S, where $\mathbf{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$. To transform the label vectors of each training instance [12] : let $\mathbf{y}_i \in \{0; 1\}^K$ be the original one-hot encoded label vector of $\mathbf{x}_i \in \mathcal{X}$ where K denotes the total number of classes, the label distribution $\mathbf{d}_i$ of $\mathbf{x}_i$ is recovered from $\mathbf{y}_i$ such that $\mathbf{d}_i \in [0; 1]^K$ and $\sum_{k=1}^K \mathbf{d}_i^k = 1$. Then, the final training set becomes $S = \{(\mathbf{x}_i, \mathbf{d}_i) \mid 1 \leq i \leq n\}$. Unlike the teacher-student method that has proven to be efficient for tasks such as music tagging [13], our approach does not rely on any deep learning-based model to infer pseudo-labels, and thus, can easily be applied on small datasets.

3. METHOD

We propose a model and task-agnostic method which aims at enhancing labels for ambiguous music classification tasks. First, a set of features is computed from each audio extract. Then, we apply a linear mapping supervised by label information that keeps training examples belonging to the same class close to each other, given a certain distance metric, and examples from distinct classes apart. We finally apply a simple label modification (i.e. label smoothing) algorithm. Our main motivation is that the original feature space might not always allow for a clear separation between the different classes, resulting in label distributions of poor quality which can later hamper training. Conceptually our approach is similar to the teacher-student method where label smoothing algorithm acts as a teacher generating more informative soft labels.

3.1. Data Representations

As a proof of concept, we use mel-frequency cepstral coefficients (MFCC) of each signal as input, as these are widely used in the music classification literature to account for timbral information [14–16]. Let us denote by $S \in \mathbb{R}^{K \times T}$ the MFCC feature matrix extracted for one 15 second long segment starting with a 5 second offset after the beginning of each audio extract, where T is the number of segments and K is the number of MFCC coefficients. We consider two strategies to aggregate the MFCCs: summarizing them through the temporal axis as the mean and standard deviation of every coefficient (MFCC-S), and a pooling and slicing strategy similar to the one introduced in [17] (MFCC-P). For the latter case, we average each MFCC coefficient over non-overlapping time windows of length M to preserve some temporal information while reducing the dimensionality. We obtain a pooled MFCC matrix $S_{\text{pooled}} \in \mathbb{R}^{K \times T'}$, where $T' = T/M$, that we then stack column-wise (e.g., appending averaged MFCC vectors for each texture window altogether). For each audio extract, we finally end up with a vector representation $\mathbf{x} \in \mathbb{R}^L$ where $L = KT'$. This representation has the advantage of carrying both timbral and temporal information while being robust against slight time-shifts, which makes it suitable for a wide range of classification tasks.

3.2. LMNN Mapping

We assume that label smoothing can better tackle label ambiguity when applied on data representations specifically suited for the classification task addressed. We propose to decompose the MFCC-P representations onto a low-dimensional space using label information. We use the Large Margin Nearest Neighbour (LMNN) algorithm [18] to learn a linear transformation A of the data that keeps the k-nearest neighbours of the same class close, while pushing examples from distinct classes apart. The algorithm relies on a given distance metric (here we use the Euclidean distance) such that the learned metric is $\tilde{d}(\mathbf{x}, \mathbf{y}) = \| A\mathbf{x} - A\mathbf{y} \|_2$. A is learned by solving the following optimization problem:

$$\min_{A \succeq 0} (1-\lambda) \sum_{(i,j) \in \mathcal{S}} d_A(\mathbf{x}_i, \mathbf{x}_j) + \lambda \sum_{(i,j,k) \in \mathcal{R}} [1 + d_A(\mathbf{x}_i, \mathbf{x}_j) - d_A(\mathbf{x}_i, \mathbf{x}_k)]_+$$

where A is some positive semi-definite matrix denoting the learned linear transformation, $\mathcal{S}$ is the set of pairs composed of $\mathbf{x}_i$ and its k-nearest neighbors of the same class, $\mathcal{R}$ is defined as a set of triples (i, j, k) such that $\mathbf{x}_i$ and $\mathbf{x}_j$ share the same label while $\mathbf{x}_k$ is a data point with a different label within the same region. Finally, $[.]_+ = \max(0, .)$ corresponds to a Hinge loss and $\lambda \in [0, 1]$ is a weighting parameter that balances the two objectives. We use the implementation of the LMNN algorithm from the metric-learn Python library [19]. The main advantage of this method is that it operates in a k-nearest neighbours (k-NN) setting, which makes it easily associable with the local k-NN smoothing algorithm.

3.3. Local k-NN Smoothing

We employ a simple k-NN approach to approximate label distributions as described in [11]. For each instance $\mathbf{x}$, its k-nearest neighbours are selected. Then, its labels' distribution is approximated by the average labels' distribution of its neighbours as follows:

$$\mathbf{d}^j = \frac{1}{k} \sum_{i \in N_k(\mathbf{x})} \mathbf{d}_i^j, (j = 1, 2, \ldots K)$$

where $N_k(\mathbf{x})$ is the index of the k nearest neighbors of $\mathbf{x}$ in the training set. In our context, the variable $\mathbf{x}$ will iterate over all the training instances of the dataset. We choose not to propagate the newly smoothed labels throughout the algorithm to avoid spreading any smoothing error. From now on, we indistinctly use label smoothing to refer to this technique.

4. EXPERIMENTS

4.1. Architectures

In order to check whether our method is influenced by the architecture, we evaluate it using two state-of-the-art CNN models for music classification:

- MusiCNN: a convolutional neural network that aims at capturing timbral and temporal patterns using vertical and horizontal convolution operations [20]. The model is composed of 6 layers and has a total of $787,000$ trainable parameters.
- VGG: originally designed for computer vision tasks, this convolutional neural network is built as a 5 layer stack of 128 3×3 convolutional filters and a has total of $605,000$ trainable parameters [21, 22].

We train both architectures targeting the original binary labels with categorical cross-entropy loss as a baseline. We additionally measure if our method remains effective in transfer learning scenarios by using public pre-trained versions of both architectures on MSD-train [23] for audio-tagging as detailed in [20, 24]. Models trained with label smoothing use a Mean Absolute Error (MAE) loss function. They are trained for the same number of epochs as their original counterparts: 150 for transfer-learning models and 600 epochs for those trained from scratch.

4.2. Datasets and evaluation

To highlight the importance of the metric learning step, we consider two versions of label smoothing for evaluation: (S_A) MFCC-S + k-NN smoothing, (S_B) MFCC-P + LMNN + k-NN smoothing.[1] We apply these two variations on the tasks detailed in Table 1: 7 mood and one genre classification tasks using small in-house MTG datasets and two publicly available music genre classification datasets: GTZAN[2] [14] and the Music Audio Benchmark Dataset from [25] (called *genre-tzanetakis* and *genre-dortmund* respectively). Note that for GTZAN, we intentionally keep the mislabelled training instances listed in [26] as they represent a typical situation of possibly corrupted and ambiguous annotations. Each model is trained in a 5-fold cross-validation setting. To measure the generalization improvement, we evaluate them on a subset of the MTG-Jamendo dataset's split-0 [27] following the methodology proposed in [24]. To avoid raters disagreement issues, we selected a subset that was identically labelled by three different annotators with labels matching the taxonomies of the considered tasks representing a few thousand tracks for each of them.[3]

The audio extracts used in this work are sampled at 44.1 kHz. MFCC (13 coefficients) are extracted using a window and FFT sizes of 1024 and 2048 respectively with a 50% overlap. For MFCC-S features, we summarize each MFCC matrix by its descriptive statistics and later apply the label smoothing method with $k_{\text{NN}} = 10$ neighbours for all tasks. For the MFCC-P features, we apply the pooling strategy on each MFCC matrix using a time window corresponding to 1 second ($M = 86$). They are then decomposed into $n = 32$ and $n = 64$ components by the LMNN algorithm for mood and genre classification respectively. We use $k_{\text{LMNN}} = 3$ neighbours and a balance parameter $\lambda = 0.5$ for the LMNN algorithm. We train it for a maximum of $10,000$ iterations. Finally, the k-NN smoothing is applied using $k_{\text{NN}} = 10$ neighbours for genre and $k_{\text{NN}} = 30$ neighbours for mood tasks.

[1] We selected the input representations that worked best for each smoothing method in pre-analysis.

[2] http://marsyas.info/downloads/datasets.html

[3] https://mtg.github.io/mtg-jamendo-dataset/annotations

Datasets	Classes	Size
genre-dortmund	alternative, blues, electronic, folk-country, funksoulrnb, jazz, pop, raphiphop, rock	1820 exc.
genre-tzanetakis	blues, classic, country, disco, hip hop, jazz, metal, pop, reggae, rock	1000 exc.
genre-rosamerica	classic, dance, hip hop, jazz, pop, rhythm and blues, rock, speech	400 ft.
mood-acoustic	acoustic, not acoustic	321 ft.
mood-electronic	electronic, not electronic	332 ft./exc.
mood-aggressive	aggressive, not aggressive	280 ft.
mood-relaxed	relaxed, not relaxed	446 ft./exc.
mood-happy	happy, not happy	302 exc.
mood-sad	sad, not sad	230 ft./exc.
mood-party	party, not party	349 exc.

Table 1: Music collections (ft.: full tracks, exc.: excerpts).

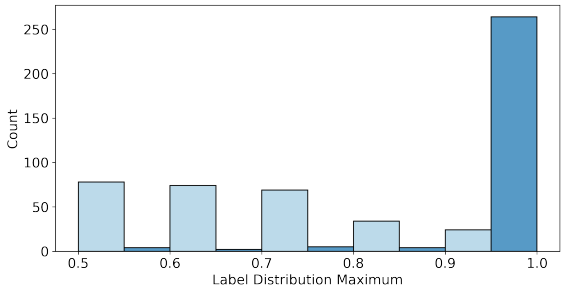

Fig. 1: Label distributions with LMNN mapping (dark blue) and without (light blue), for the *mood-electronic* dataset.

5. DISCUSSION

Figure 1 shows the overall distribution of the newly constructed labels in a low-separability case. For the S_B approach, the resulting distributions are skewed towards 1, which indicates that most songs are well anchored in their respective class. On the contrary, label distributions are centered around 0.65[4] when following the S_A approach, which indicates that the feature space implied by MFCC-S seems to be fuzzier without any clear separation among classes. The characteristics of these distributions and subjective listening evaluations led us to believe that the LMNN-based transformation manages to conveniently combine acoustic attributes contained in pooled MFCCs to spread data points onto a space that is more faithful to class information. The more neighbours are used by the k-NN smoothing, the more "fine-grained" the resulting label distributions will be.

[4] Other experiments using the deep embeddings from the MusiCNN pre-trained model have also led to similar results.

Architecture	Dataset	Trained from scratch						Pre-trained on MSD					
		5F			Ext.			5F			Ext.		
		None	S_A	S_B	None	S_A	S_B	None	S_A	S_B	None	S_A	S_B
MusiCNN	genre-dortmund	0.42	0.29	0.36	0.44	0.38	0.41	0.54	0.46	0.49	0.53	0.46	0.52
	genre-rosamerica	0.85	0.82	0.83	0.65	0.51	0.59	0.86	0.85	0.83	0.73	0.58	0.69
	genre-tzanetakis	0.83	0.77	0.81	0.52	0.31	0.47	0.79	0.76	0.77	0.59	0.46	0.56
	mood-acoustic	0.91	0.92	0.91	0.78	0.77	**0.79**	0.93	0.92	0.92	0.81	**0.84**	**0.86**
	mood-electronic	0.82	0.82	0.85	0.77	**0.78**	0.77	0.91	0.86	0.87	0.79	**0.81**	**0.85**
	mood-aggressive	0.96	0.95	0.96	0.73	**0.79**	**0.80**	0.98	0.97	0.97	0.84	**0.85**	0.84
	mood-relaxed	0.89	0.88	0.88	0.77	0.76	**0.79**	0.87	0.88	0.90	0.77	0.73	**0.78**
	mood-happy	0.72	0.76	0.77	0.60	0.59	**0.61**	0.86	0.82	0.85	0.63	0.63	0.63
	mood-sad	0.84	0.82	0.84	0.65	0.64	0.60	0.86	0.85	0.87	0.71	0.70	0.70
	mood-party	0.90	0.89	0.88	0.80	0.79	**0.82**	0.90	0.91	0.91	0.84	**0.85**	**0.85**
VGG	genre-dortmund	0.43	0.36	0.41	0.44	0.36	0.38	0.33	0.29	0.31	0.36	0.29	0.31
	genre-rosamerica	0.87	0.81	0.85	0.65	0.51	0.65	0.64	0.65	0.66	0.50	0.47	**0.66**
	genre-tzanetakis	0.82	0.80	0.82	0.52	0.49	**0.59**	0.64	0.67	0.64	0.64	0.27	0.64
	mood-acoustic	0.89	0.94	0.94	0.80	0.77	0.80	0.90	0.91	0.89	0.80	0.79	**0.83**
	mood-electronic	0.83	0.78	0.82	0.79	0.78	**0.80**	0.89	0.80	0.87	0.79	0.79	**0.84**
	mood-aggressive	0.97	0.96	0.97	0.75	**0.80**	0.75	0.93	0.93	0.92	0.81	**0.83**	0.81
	mood-relaxed	0.83	0.88	0.90	0.80	0.78	0.77	0.84	0.84	0.87	0.67	**0.68**	**0.68**
	mood-happy	0.82	0.79	0.78	0.67	0.63	0.65	0.83	0.82	0.82	0.61	0.61	0.61
	mood-sad	0.86	0.85	0.88	0.69	0.66	**0.70**	0.83	0.83	0.83	0.65	0.63	0.62
	mood-party	0.90	0.90	0.91	0.81	0.79	0.79	0.88	0.88	0.89	0.78	0.77	**0.79**

Table 2: Evaluation results. 5F: 5-fold cross-validation results, statistically significant differences over baseline according to an independent samples t-test ($P > 0.05$) are marked in light grey. Ext.: External validation results (class-weighted accuracies), no worse than the baseline are marked in grey, higher accuracies are marked in bold. None: No smoothing applied (baseline).

In a small dataset scenario, the capacity for a model to generalize to unseen data is usually limited [24]. However, results reported in Table 2 on the external audio collection show that the label distribution learning paradigm helps to make the model more robust against data proceeding from different distributions. Even though the parameters used for our smoothing method remained the same within each dataset's category, each model's performance either improves or does not degrade in most classification tasks. By acting as an implicit label augmentation, label distribution learning makes the classification task more complex and reduces the negative influence of ambiguous and mislabelled training data points. Even though both approaches lead to consistent improvements over the baselines, the S_B method tends to outperform the S_A one in most cases. However, the *mood-aggressive* task seems to be straightforward enough to be efficiently improved only using the S_A approach. Additionally, the generalization improvement does not seem to directly depend neither on the architecture used, nor the training strategy followed, which makes our approach suitable for a wide variety of models and classification tasks involving ambiguous categories. Further improvements can be made with a simple parameter search per classification task. For example, datasets with more than two classes seem to require a higher LMNN output dimension and a smaller number of neighbours used during the smoothing step. Finally, results on pre-trained models show that our approach can be used to further enhance knowledge transfer among source and target tasks in transfer learning scenarios.

6. CONCLUSION

We have introduced a method to explicitly model ambiguity arising in various music classification tasks. Our method shows that carefully transforming labels can help limiting overfitting in small datasets scenarios and ultimately improve generalization to out-of-sample data. We experimented with two state-of-the-art CNN architectures and a variety of music classification tasks. In most cases, our method improved generalization without degrading 5-fold cross-validation results. It also remains compatible with pre-trained models, and seems to ameliorate transfer among tasks. We argue that finding perceptually relevant low-dimensional spaces can greatly boost the impact of label smoothing approaches. To better approximate perceptual ambiguity, we suggest modifying the triplet selection mechanism of the LMNN algorithm to make it more robust to mislabelled data points. Finally, it would be interesting to measure the effectiveness of our approach on other MIR tasks indirectly involving classification such as instrument recognition or music structure analysis.

7. REFERENCES

[1] Sander Dieleman and Benjamin Schrauwen, "End-to-end learning for music audio," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, 2014.

[2] Jordi Pons, Thomas Lidy, and Xavier Serra, "Experimenting with musically motivated convolutional neural networks," in *International Workshop on Content-based Multimedia Indexing (CBMI 2016)*, 2016.

[3] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho, "Convolutional recurrent neural networks for music classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017)*, 2017.

[4] Xin Geng, Chao Yin, and Zhi-Hua Zhou, "Facial age estimation by learning from label distributions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 10, pp. 2401–2412, 2013.

[5] Seong G Kong and Ralph Oyini Mbouna, "Head pose estimation from a 2D face image using 3D face morphing with depth parameters," *IEEE Transactions on Image Processing*, vol. 24, no. 6, pp. 1801–1808, 2015.

[6] Bin-Bin Gao, Chao Xing, Chen-Wei Xie, Jianxin Wu, and Xin Geng, "Deep label distribution learning with label ambiguity," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2825–2838, 2017.

[7] Youngmoo E Kim, Erik M Schmidt, Raymond Migneco, Brandon G Morton, Patrick Richardson, Jeffrey Scott, Jacquelin A Speck, and Douglas Turnbull, "Music emotion recognition: A state of the art review," in *Proceedings of the 11th International Society for Music Information Retrieval Conference*, 2010, vol. 86, pp. 937–952.

[8] Cory McKay and Ichiro Fujinaga, "Musical genre classification: Is it worth pursuing and how can it be improved?," in *International Society for Music Information Retrieval Conference (ISMIR 2006)*, 2006.

[9] Juan Sebastián Gómez Cañón, Estefanía Cano, Herrera Boyer, Emilia Gómez Gutiérrez, et al., "Joyful for you and tender for us: The influence of individual characteristics and language on emotion labeling and classification," in *Proceedings of the 21st International Society for Music Information Retrieval Conference; 2020 Oct 11-16; Montréal, Canada*, 2020.

[10] Jean-Julien Aucouturier, François Pachet, Pierre Roy, and Anthony Beurivé, "Signal+ context = better classification.," in *International Society for Music Information Retrieval Conference (ISMIR 2007)*, 2007.

[11] Xin Geng, "Label distribution learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, 2016.

[12] Ning Xu, Yun-Peng Liu, and Xin Geng, "Label enhancement for label distribution learning," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[13] Minz Won, Keunwoo Choi, and Xavier Serra, "Semi-supervised music tagging transformer," in *International Society for Music Information Retrieval Conference (ISMIR 2021)*, 2021.

[14] George Tzanetakis and Perry Cook, "Musical genre classification of audio signals," *IEEE Transactions on speech and audio processing*, vol. 10, no. 5, pp. 293–302, 2002.

[15] Nicolas Scaringella, Giorgio Zoia, and Daniel Mlynek, "Automatic genre classification of music content: a survey," *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 133–141, 2006.

[16] Monica S Nagawade and Varsha R Ratnaparkhe, "Musical instrument identification using MFCC," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 2017.

[17] Nicolas Scaringella and Giorgio Zoia, "On the modeling of time information for automatic genre recognition systems in audio signals.," in *International Society for Music Information Retrieval Conference (ISMIR 2005)*, 2005.

[18] Kilian Q Weinberger and Lawrence K Saul, "Distance metric learning for large margin nearest neighbor classification.," *Journal of machine learning research*, vol. 10, no. 2, 2009.

[19] William de Vazelhes, CJ Carey, Yuan Tang, Nathalie Vauquier, and Aurélien Bellet, "metric-learn: Metric Learning Algorithms in Python," *Journal of Machine Learning Research*, vol. 21, no. 138, pp. 1–6, 2020.

[20] Jordi Pons and Xavier Serra, "musicnn: Pre-trained convolutional neural networks for music audio tagging," *International Society for Music Information Retrieval Conference (ISMIR 2019) Late-Braeking/Demo*, 2019.

[21] Keunwoo Choi, Gyorgy Fazekas, and Mark Sandler, "Automatic tagging using deep convolutional neural networks," in *International Society for Music Information Retrieval Conference (ISMIR 2016)*, 2016.

[22] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al., "Cnn architectures for large-scale audio classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017)*, 2017.

[23] Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whitman, and Paul Lamere, "The Million Song Dataset," 2011.

[24] Pablo Alonso-Jiménez, Dmitry Bogdanov, Jordi Pons, and Xavier Serra, "TensorFlow audio models in Essentia," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*, 2020.

[25] Helge Homburg, Ingo Mierswa, Bülent Möller, Katharina Morik, and Michael Wurst, "A benchmark dataset for audio classification and clustering.," in *International Society for Music Information Retrieval Conference (ISMIR 2005)*, 2005.

[26] Bob L Sturm, "An analysis of the gtzan music genre dataset," in *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, 2012, pp. 7–12.

[27] Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alastair Porter, and Xavier Serra, "The MTG-Jamendo Dataset for automatic music tagging," in *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML 2019)*, 2019.