A COMBINATION OF LOCAL APPROACHES FOR HIERARCHICAL MUSIC GENRE CLASSIFICATION

Antonio R. S. Parmezan[1] **Diego Furtado Silva**[2] **Gustavo E. A. P. A. Batista**[3]

[1] Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo, São Carlos, Brazil

[2] Departamento de Computação – Universidade Federal de São Carlos, São Carlos, Brazil

[3] School of Computer Science and Engineering – University of New South Wales, Sydney, Australia

parmezan@usp.br, diegofs@ufscar.br, g.batista@unsw.edu.au

ABSTRACT

Labeling a music recording according to its genre is an intuitive and familiar way to describe its content. Music genres are valuable information especially for music organization, personalized listening experience, and playlist generation. Automatically classifying music genres is a challenging endeavor due to the inherent ambiguity and subjectivity. Most efforts on music genre classification consider the complete independence between labels. However, music genres typically respect a hierarchical structure based on the influences or origins of each style. Conversely, many of the methods available for hierarchical classification are based on assumptions about the class hierarchy, such as the need for multiple children in each hierarchy's node, which may limit their use in music applications. Also, the local classifier per node approach that would be the most suitable for this scenario is costly regarding time and memory. In this paper, we present two local hierarchical classification approaches and show how to combine them to obtain a single one that is more robust and faithful to the music genre classification scenario. We evaluate our proposal in a music dataset hierarchically labeled with 120 music genres. As shown, compared to state-of-the-art approaches, our approach has a lower computational cost and can achieve competitive performances.

1. INTRODUCTION

The music genre is a convention used by humans to categorize and organize pieces of music. Besides being essential metadata for large databases of music distribution, the music genre resides in one of the most common descriptors employed in studies involving storage, retrieval, and usage of music knowledge [1–3].

The major problem with music genre information is that it is usually fuzzy and inaccurate. At the core of this question is human subjectivity, closely related to the several criteria for labeling music in genres [4]. Because of subjectivity, the music genre classification task becomes even more challenging, since it needs to deal with differences in interpretation and an unavoidable intersection between genres.

Besides, as there is no standard for labeling music, some music platforms may present a high number of genres. For example, the streaming music service Spotify [1] catalogs its music in over 1500 genres, which will eventually be updated and increased in quantity in the future. Also, a song classified as, for instance, "gothic metal" on Spotify may be labeled as "alternative rock" and "indie rock" on other platforms, such as Google Play Music [2] and Apple Music [3]. Finally, some music genres overlap in these services, while others have subgenres. Although there is no unique way of determining an item's music genre, the literature covers distinct approaches and methods to support this complex and many-sided task.

So far, most of the work in music information retrieval is only concerned with music genres as a flat classification problem [5–8]. A flat classifier seeks to associate each example with a class that belongs to a finite, devoid of structural dependencies and usually small, set of classes. However, the music genre classification calls for a genre taxonomy, *i.e.*, a hierarchical set of categories to be mapped onto a music collection.

In hierarchical music genre classification, supervised machine learning algorithms are designed to induce a hierarchical decision model. Such a model links the features of the examples to a class hierarchy, generally represented as a tree or a direct acyclic graph with varying specificity and generality levels. An advantage in assigning examples to hierarchically organized classes is that the closer to the root of the hierarchy a linkage occurs, the smaller the classifier error rate tends to be. Conversely, the obtained classification will be less specific and, therefore, less informative [9, 10].

Having the genres structured into a class hierarchy helps users to not only browse and retrieve music pieces but also navigate the collection according to the similarity of its content.

Most of the methods available for hierarchical classification are based on the local classifier per node approach.

[1] https://www.spotify.com.

[2] https://play.google.com/music.

[3] https://www.apple.com/music.

This approach resides in training a binary classifier for each node of the class hierarchy, except the root node, aiming to predict whether or not an example has the corresponding class [11]. The local binary dataset for a given node in the class hierarchy contains positive examples, those that have some relation to the class of the node in question, and negative examples, those related to the remaining classes. We can determine the sets of positive and negative examples in different ways; for instance, adopting heuristics based on set operations [12] or strategies based on nearest neighbors [13]. Depending on the number and disposition of the classes in the hierarchy, we may choose to use multiclass classifiers instead of binary ones. Thus, while the local classifier per parent node approach builds a multiclass classifier for each non-leaf class node of the hierarchy, the local classifier per level approach generates a multiclass classifier for each hierarchy level [11].

The multiclass approaches are much more efficient concerning time and memory than the binary one because they build fewer classifiers. On the other hand, they make some assumptions about the class hierarchy, such as the need for multiple children in each hierarchy's node. Although the assumptions help to avoid both the generation of one-class local datasets and class-membership inconsistency, they may limit the direct use of multiclass classifier-based hierarchical approaches in music applications.

In this paper, we combine the per node and per parent node approaches to obtain a single local hierarchical method that is more robust and faithful to the music genre classification scenario. We advocate using decision tree-based classification algorithms to build the local classifiers because they internally perform feature selection. As discussed throughout this work, this is a relevant consideration in hierarchical music genre classification since the features that most distinguish among classes tend to be different at each level of the class hierarchy.

We evaluate the proposed approach using a music dataset with 120 hierarchically organized music genres. Also, we compare our proposal with three well-known approaches in terms of performance and runtime. The results show that our proposal is computationally inexpensive and competitive against the traditional approaches.

The remainder of this paper is organized as follows: Section 2 gives a general view of the problem and summarizes research efforts in automatic music genre classification; Section 3 introduces the concepts of hierarchical classification; Section 4 describes our proposal, from the design of the hierarchical decision model to the prediction strategy adopted; Section 5 presents the experimental protocol, as well as the results and our discussion about them; Section 6 concludes this study and shows directions for future work.

2. RELATED WORK

Recognition of music genres is one of the most prominent research problems in music information retrieval. Studies point out that genre is the most chosen concept to guide the user browsing in music repositories [1, 14].

A music genre recognition system aims to categorize an audio signal with an unknown label into a previously known music genre from relevant features extracted from this audio. A classifier makes use of such characteristics to identify the music genre of the analyzed signal. The benefits of categorizing pieces of music in genres extend to many other tasks, such as organizing digital audio databases [15], building new search engines [16], and recommending songs [17].

Several approaches for audio-based music genre classification extract features associated with the timbral information, such as Mel-Frequency Cepstral Coefficients (MFCC), and sets of spectral and rhythmic characteristics [18].

There are three main categories of algorithms that can be applied in this context [19]. The first one represents the entire recording with one single set of features [5, 20]. The other two rely on classifying feature vectors extracted from short frames of the recordings and achieve the final result by an ensemble of these classifications. Some methods perform this approach directly on the frame-level features [21], while some other aggregate few consecutive frames, creating a new set of features usually given by the mean and standard deviations of the aggregated features [6, 22, 23].

More recently, researchers have used deep learning models to learn a feature representation that promises to advance the task of genre recognition significantly [24–27]. Although deep learning methods are computationally expensive, they allow the extraction of relevant audio features without having to depend on ad-hoc domain-dependent signal processing strategies [24].

The literature has mostly treated the automatic music genre classification as a problem of flat classification. In other words, most papers in this domain consider all the genres in the same hierarchy's level [5–8]. However, the music genre classification problem is better modeled with a taxonomy of genres. We describe below some notable works that have used class hierarchies to support the task mentioned above.

A binary approach, which uses a feature selection method on each generated local dataset and a Gaussian Mixture Model as a base-level classifier, was proposed in [28]. In this same direction, the authors in [29] applied an ensemble of Feed Forward Neural Networks and k-Nearest Neighbors (kNN) over the local binary datasets. For the kNN classifier, they employed a genetic algorithm feature selection mechanism.

Two datasets with content-based features and a standard local approach using Support Vector Machines classifiers were explored in [30]. Differently, the authors in [31] developed a local approach that adopts feature selection, multiple representations from the same object, and enables hierarchically multi-label classifications by using a two-layer labeling process.

The study reported in [32] involved two selective multiclass hierarchical methods. The first one selects the best feature set instead of the best classifier, while the second one selects both the best classifier and the best representation simultaneously.

The authors in [33] proposed a novel approach to building a classification tree through subspace cluster analysis. On the other hand, hierarchical analysis of spectrograms was investigated in [34] to help classify music in genres.

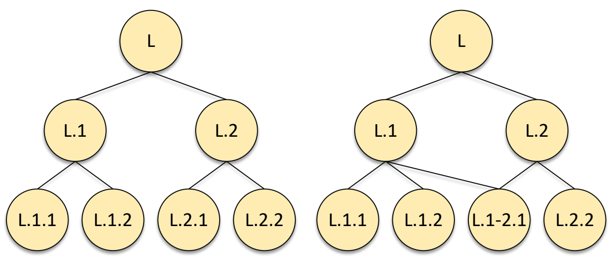
This paper presents a proposal that differs from the literature for having another point of view. Here we combine two local approaches to quickly and efficiently obtain a single hierarchical method that faithfully represents the music genre classification scenario.

3. HIERARCHICAL CLASSIFICATION

Flat classification differs from hierarchical one because in the latter the domain classes follow a logical organization. In flat classification, while the absence of interrelationships between classes characterizes some problems (single-label classification), the non-structural relationships between labels evidence others (multi-label classification). Structural dependencies, which express super or subclass relations, define hierarchical classification.

A dataset for hierarchical classification in the attribute-value table format comprises N pairs of examples $(\vec{x}_i, Y_i)$, where $\vec{x}_i = (x_{i_1}, x_{i_2}, x_{i_3}, \ldots, x_{i_M})$ and $Y_i \subset L = \{\text{L}, \text{L.1}, \text{L.1.1}, \text{L.1.2}, \ldots\}$. Specifically, each example $\vec{x}_i$ is represented by M predictive features (attributes) and has a set of labels Y_i for which there are relationships that obey a hierarchical class structure stipulated *a priori*. The class attribute, in turn, reflects the concept to be learned and described by the induced hierarchical models using supervised machine learning algorithms.

We can discern the hierarchical classification methods according to four central aspects [11]. The first one covers the type of hierarchical structure – tree or Direct Acyclic Graph (DAG) –, taken to depict the relationships among classes. In the tree structure (Figure 1(a)), each node, except the root node, is linked with at most one parent node. In the DAG structure (Figure 1(b)), each node, except the root node, can have one or more parent nodes.



(a) Tree (b) DAG

Figure 1. Hierarchical class structures.

The second aspect determines whether the algorithm can predict classes in one or more paths in the hierarchical structure. For instance, in the class hierarchy tree of Figure 1(a), if the model is able to predict both classes L.1.1 and L.1.2 for a provided example, which refers to the paths L→L.1→L.1.1 and L→L.1→L.1.2, then it can predict multiple paths – Multiple Path Prediction (MPP). Conversely, the method performs Single Path Prediction (SPP) when this type of association is invalid.

The third aspect involves the hierarchical level at which the classification takes place. An algorithm can predict using only classes represented by leaf nodes – Mandatory Leaf-Node Prediction (MLNP) – or by using classes denoted by any internal or leaf node within the hierarchical structure – Non-Mandatory Leaf-Node Prediction (NMLNP). Figure 2 illustrates these two prediction strategies; the path L→L.2→L.2.1 portrays the NMLNP strategy, and the path L→L.2→L.2.1→L.2.1.3 indicates the MLNP tactic. We need to highlight that the NMLNP strategy is convenient mainly in applications that opt for the freedom to conduct a more generic classification, but with greater reliability.

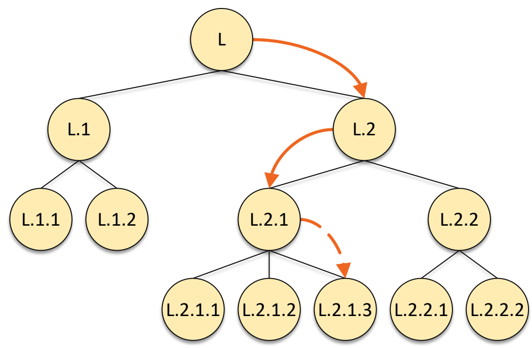


Figure 2. Hypothetical class hierarchy.

The fourth and final aspect concerns the way that machine learning methods deal with the hierarchical structure. We can group the approaches described in the literature into three broad categories: (i) flat approach, (ii) local approach, and (iii) global approach. Further details are available in [11].

4. PROPOSED APPROACH

One interesting research aspect that has been neglected by the music information retrieval and machine learning communities is the development and evaluation of the fusion of two or more hierarchical classification approaches. The motivation behind this idea arises because, unlike the global approach that generates a single classifier whose structure includes the entire class hierarchy [32, 35, 36], traditional approaches work with several local classifiers – binary or multiclass – to model the taxonomy of the problem's labels [9, 11].

Local approaches are generally preferred over the global ones due to the possibility of employing conventional supervised machine learning algorithms, which have been extensively tested and validated in flat classification tasks [37]. However, while some of the local approaches are computationally expensive in terms of time and memory, others make assumptions about the class hierarchy and, as a result, cannot be directly applied in some scenarios like the one treated here.

As aforementioned, the purpose of this paper is to combine the per node and per parent node local approaches to obtain a more efficient one. The justification for proposing a hybrid approach comes from the music genre classification problem, and we will explain it below using the class hierarchy tree of Figure 3.

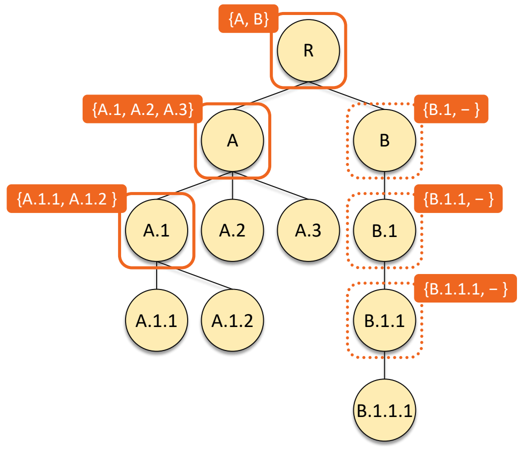


Figure 3. Proposed approach: a hybrid local approach based on the per node and per parent node approaches. Squares with curved corners symbolize multiclass classifiers. Squares with dotted curved corners depict binary classifiers.

The class hierarchy of Figure 3 shows 11 classes that can be interpreted as music genres. A music genre can have one or more subgenres. If we adopted the local classifier per node approach, we would create a binary classifier for each class in the hierarchy, except for the root node, employing a set of positive examples – examples representing the current class – and a set of negative examples – examples that they are not associated with the current class. In this sense, to find the local training sets related to each class from the training dataset, several heuristics have been proposed [12, 13, 37]. The per node approach is suited to the task of music genre classification, but due to the construction of $|L| - 1$ classifiers, it is expensive regarding memory and processing.

On the other hand, if we were to use the local classifier per parent node approach, we would build, for each non-leaf node in the class hierarchy, a multiclass classifier to label new examples according to their subclasses. Therefore, in this approach, classifiers are also generated from sets of local training examples. Each local training set should be prepared so that the examples included therein are labeled only with the classes that will be differentiated by the multiclass classifier. The label of each example inserted in the training set selected for the classifier's induction must be generalized so that only the labels referring to the child classes of the analyzed node are present. Here a problem arises: if the approach were to encounter nodes B, B.1, and B.1.1 in Figure 3, it would generate one-class local datasets. Single paths like B→B.1→B.1.1→B.1.1.1 are essential in music genre classification since it demonstrates the evolution of the genre over time. Despite this issue, the per parent node approach has better memory and processing than the per node one, and the resulting classifiers are less complex than the per level approach.

The local classifier per level approach, which creates a multiclass classifier for each level of the hierarchy, is not exempt from limitations. For the fourth level of the tree structure in Figure 3, such an approach would generate a local dataset containing only one class (B.1.1.).

In order to address the above issues, we propose a hybrid approach that represents the class hierarchy as a tree. We set up it as follows: for each internal node in the hierarchy with two or more children, we apply the local classifier per parent node approach and build a multiclass classifier with the children nodes, as indicated by the squares with curved corners in Figure 3. Otherwise, if the node has only one child, we apply the local classifier per node approach and generate a binary classifier, as symbolized by the squares with dotted curved corners in Figure 3. Note that we need to employ a strategy to choose the positive and negative examples from the local binary datasets. The literature shows that the more examples we consider in the learning phase, the better the induced classifier performs [12].

In this work, we suggest applying the "inclusive" heuristic to create local datasets [12]. This heuristic defines that the set of positive examples (S^+) is the most specific class and its descendants. In contrast, the set of negative examples (S^-) is all other classes except those in the set of positive examples and the ancestors of the most specific class. The output of the "inclusive" heuristic for node L.2.1 is: $S^+ = \{$L.2.1, L.2.1.1, L.2.1.2, L.2.1.3$\}$, and $S^- = \{$L.1, L.1.1, L.1.2, L.2.2, L.2.2.1, L.2.2.2$\}$.

At each level of the hierarchy, groups of music genres are distinguished by their differences. Intuitively, the acoustic features that discriminate gothic metal from pop music diverge from those that differentiate country from gospel music. Hence, we can affirm that the classification of distinct music genres, like many other objects, benefits from different representations at distinct levels of the hierarchy. For this reason, we advocate applying learning algorithms based on decision trees to induce the local classifiers since they internally perform feature selection.

Finally, the hierarchy of music genres has some peculiarities that allow the classification to stop at internal nodes or go down up to the leaf nodes. To avoid inconsistencies in the classification step, we recommend using the top-down prediction strategy. In this strategy, an example is initially classified – based on the classifier's reliability (classification score) – among the first-level classes, and the subtrees of interest are only used to classify the examples at the other levels. The classification procedure is interrupted when the current classifier's reliability is less than a predefined $threshold$.

5. EXPERIMENTAL EVALUATION

This section presents an empirical assessment of our proposal and its comparison with three well-known approaches (Figure 4). First, we describe the considered dataset. Next, we report the experimental setup regarding the adversary approaches, parameter setting, and evaluation measures. Then, we show and discuss the obtained results in terms of predictive performance and learning time.

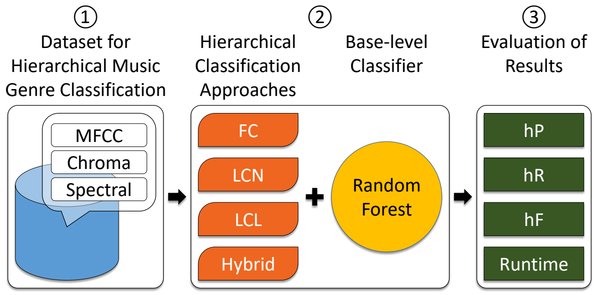


Figure 4. High-level overview of the empirical evaluation.

5.1 Dataset

In our experiments, we used data from the Free Music Archive (FMA) [38]. This dataset contains 106,574 recordings, organized in 161 imbalanced genres.

We note that this paper's focus is not proposing novel features or comparing them in the classification scenario. For this reason, we considered the features available with the FMA dataset instead of extracting or learning features from the audios. Such a decision allowed us to compare the hierarchical classification approaches without relying on the gain provided by a better (or worse) feature set.

The features provided with the dataset, which we applied in our computational tests, are statistics from windowed MFCC, chroma, and spectral features extracted from 30 seconds in the middle of each recording using the LibROSA framework [39]. Specifically, these statistics are the mean, standard deviation, skew, kurtosis, median, minimum, and maximum.

As noted in Section 2, music genre classification is usually performed by timbre-related features, such as MFCC and spectral characteristics. However, we also included the chroma-based features in our experiments to evaluate both the contribution of this kind of feature in the genre classification and our proposal's behavior when dealing with distinct music characteristics.

The FMA dataset has a default training-validation-test split. As we did not use the validation set to tune parameters, we merged it with the test examples. This operation provided us a training set that comprises 80% of the examples. Consequently, 20% of the remaining recordings belong to the test partition. Besides, this split is stratified and is guaranteed that there is no artist in the test set that also appears in the training set.

The genre hierarchy is an interesting characteristic to observe in this dataset. As the FMA allows the artists to label their songs themselves, the dataset presents a complex hierarchy of genres.

Another challenging factor present in this dataset is the class imbalance – "unbalanced with 1 to 38,154 tracks per genre" [38].

Figure 5 illustrates one branch of the class hierarchy, which presents the stated issues. It has nodes with single and multiple children, as well as genres with a significant difference regarding the number of tracks.

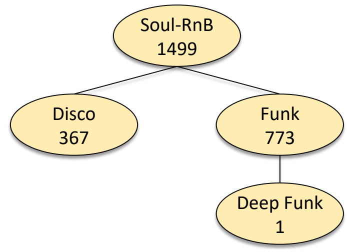


Figure 5. Example of genre hierarchy for the top-level Soul-RnB genre.

To make the hierarchical classification of single paths feasible, we address some issues found in the FMA dataset. In so many cases, the associated genres belong to more than one path in the genre hierarchy tree. In these cases, we select the genre for which the leaf node is at the lowest level. In case of ties, we kept that one in which the genre in the lowest level of each path comprises the higher number of tracks. We also removed unlabeled examples and examples from the test set whose classes were not present in the training set. Therefore, the dataset evaluated in this paper has 82,374 training examples and 15,681 test examples. These examples are described by 461 features and are associated with 120 classes organized hierarchically. The class hierarchy has four levels, each with 21, 107, 17, and 2 classes.

5.2 Compared Approaches and Parameter Settings

We compared the hybrid approach with three other well-known ones: (i) Flat Classifier (FC), (ii) Local Classifier per Node (LCN), and (iii) Local Classifier per Level (LCL). For FC, we assume each possible path in the label tree to be a class. For LCN, we use the "inclusive" heuristic to generate local datasets.

We adopted Random Forest [40] as a symbolic base-level classifier with the number of variables available for splitting at each tree node ($mtry$) equal to $\sqrt{(M-1)}$.

We considered the NMLNP strategy with $threshold = 0.5$. This setting means that the classification at the deepest levels is interrupted when the classification score is less than 0.5 or, in the case of per node classifiers, the predicted class is negative.

The experimental protocol execution comprised the use of the programming language R [4] with the following packages: caret, data.table, data.tree, and doParallel.

5.3 Evaluation Measures

In hierarchical classification problems, classes belonging to the levels furthest from the root node are generally more difficult to predict than classes associated with levels closest to the root node. In view of this, we assessed the quality

[4] `https://www.r-project.org`.

of the hierarchical classification approaches according to three hierarchical performance measures: (i) hierarchical Precision (hP), (ii) hierarchical Recall (hR), and (iii) hierarchical F-measure (hF). They are defined as follows:

$$hP = \frac{\sum_i |Y_i \cap \hat{Y}_i|}{\sum_i |\hat{Y}_i|} \quad (1) \qquad hR = \frac{\sum_i |Y_i \cap \hat{Y}_i|}{\sum_i |Y_i|} \quad (2)$$

$$hF = \frac{(\beta^2 + 1) \times hP \times hR}{\beta^2 \times hP + hR} \quad (3)$$

In Eqns. 1 and 2, $\hat{Y}_i$ denotes the set of labels predicted for a test example i and Y_i corresponds to the set of classes correct for this example. During the *hP* and *hR* computations, we need to discard the root node of the label hierarchy since, by definition, it is common to all the examples. The summations, in turn, are calculated over all the test examples.

As for Eqn. 3, β belongs to $[0, \infty)$ and refers to the importance assigned to the hP and hR values. Here, we established $\beta = 1$.

We must emphasize that the described measures are extended versions of the well-known metrics of precision, recall, and F-measure but tailored to the hierarchical classification scenario.

5.4 Results and Discussion

In order to check the performance of the hybrid method, we made comparisons with three other hierarchical classification approaches: FC, LCN, and LCL.

Our first evaluation criterion to be analyzed is predictive performance. Table 1 exhibits the three hierarchical performance measures obtained in the FMA dataset for the proposed method, as well as for the approaches used as baselines.

	Hierarchical approaches			
Performance measure	FC	LCN	LCL	Hybrid
hP	67.62	72.36	69.86	75.79
hR	66.10	71.96	69.46	77.38
hF	66.88	72.16	69.66	76.57

Table 1. Hierarchical predictive performance in % of the traditional approaches from the literature compared to our approach.

As shown in Table 1, the hybrid approach provided the best results, surpassing by a margin of approximately 4% the LCN scheme that is widely applied in the related literature. The poorest results came from the flat classifier. We expected this since such a model completely ignores the problem class hierarchy and, consequently, does not use domain knowledge to decompose the feature space of the problem in question into subproblems with a smaller number of classes.

After the predictive performance, our second evaluation criterion is runtime. In this work, runtime refers to learning time, *i.e.*, the time spent in inducing hierarchical classifiers over a dataset. Table 2 presents the runtime results achieved in the FMA dataset using both our proposal and the baseline approaches. We performed the tests on a server running 2.10 GHz Intel Xeon E5-2620 v4 processor (32-core) with 92GB RAM and operational system Debian 4.9.130-2 (64 bits) under the same processing conditions for all measurements. The times are indicated in minutes (min) or hours (h).

	Hierarchical approaches			
	FC	LCN	LCL	Hybrid
Learning time	48 min	92.44 h	4.41 h	3.53 h

Table 2. Time costs of the traditional approaches from the literature compared to our approach.

In Table 2, we can see that the hybrid method had a relatively shorter execution time than the LCN and LCL approaches. While LCN built 119 binary models and LCL four multiclass classifiers, our approach induced 27 models, 16 of which are multiclass and 11 binaries. We highlight that although our proposal has generated more classifiers than LCL, the induced models involved fewer examples and classes.

Even though the flat classifier's learning time was shorter than that of the hybrid method, the latter provided the best results in terms of predictive performance.

6. CONCLUDING REMARKS

In this paper, we introduced a novel approach for hierarchical music genre classification. The proposed method is based on combining local approaches to adapt the genre classification to more realistic hierarchies of music genre. Our results showed that the designed approach is better than the traditional ones in terms of predictive performance and execution time.

As future work, we intend to extend our method to deal with multiple labels in distinct paths in the hierarchy. Also, we plan to evaluate the use of different features and learning algorithms for the local classifications.

Finally, this research covered one real music genre hierarchy. As mentioned, distinct music platforms organize their labels in different ways. Thus, it is also our interest to perform a broad study on other real music genre structures to find specific contrivances that may aggregate information to improve hierarchical classifications in the context of music data.

7. ACKNOWLEDGEMENT

This work was financed in part by the Brazilian National Council for Scientific and Technological Development [grant numbers 140159/2017-7 and 142050/2019-9], and the São Paulo Research Foundation [grant number 2017/24340-6].

8. REFERENCES

[1] J.-J. Aucouturier and F. Pachet, "Representing musical genre: a state of the art," *Journal of new music research*, vol. 32, no. 1, pp. 83–93, 2003.

[2] E. Pampalk and M. Goto, "MusicSun: a new approach to artist recommendation," in *International Society for Music Information Retrieval Conference*, 2007, pp. 101–104.

[3] D. C. Corrêa and F. A. Rodrigues, "A survey on symbolic data-based music genre classification," *Expert Systems with Applications*, vol. 60, pp. 190–210, 2016.

[4] S. Lippens, J.-P. Martens, and T. De Mulder, "A comparison of human and automatic musical genre classification," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4. IEEE, 2004, pp. iv–iv.

[5] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[6] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kégl, "Aggregate features and ADABOOST for music classification," *Machine Learning*, vol. 65, no. 2-3, pp. 473–484, 2006.

[7] A. Holzapfel and Y. Stylianou, "Musical genre classification using nonnegative matrix factorization-based features," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 424–434, 2008.

[8] I. Panagakis, E. Benetos, and C. Kotropoulos, "Music genre classification: a multilinear approach," in *International Society for Music Information Retrieval Conference*, 2008, pp. 583–588.

[9] J. G. Colonna, J. Gama, and E. F. Nakamura, "A comparison of hierarchical multi-output recognition approaches for anuran classification," *Machine Learning*, vol. 107, no. 11, pp. 1651–1671, 2018.

[10] A. R. S. Parmezan, V. M. A. Souza, and G. E. A. P. A. Batista, "Towards hierarchical classification of data streams," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, ser. Lecture Notes in Computer Science. Springer, 2019, vol. 11401, pp. 314–322, cIARP 2018.

[11] C. N. Silla and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining and Knowledge Discovery*, vol. 22, no. 1, pp. 31–72, 2011.

[12] R. Eisner, B. Poulin, D. Szafron, P. Lu, and R. Greiner, "Improving protein function prediction using the hierarchical structure of the gene ontology," in *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*. IEEE, 2005, pp. 1–10.

[13] T. Fagni and F. Sebastiani, "On the selection of negative examples for hierarchical text categorization," in *Language & Technology Conference*, 2007, pp. 24–28.

[14] J. H. Lee, H. Cho, and Y.-S. Kim, "Users' music information needs and behaviors: Design implications for music information retrieval systems," *Journal of the Association for Information Science and Technology*, vol. 67, no. 6, pp. 1301–1330, 2016.

[15] M. Torrens, P. Hertzog, and J. L. Arcos, "Visualizing and exploring personal music libraries," in *International Society for Music Information Retrieval Conference*, 2004.

[16] P. Knees, T. Pohle, M. Schedl, and G. Widmer, "A music search engine built upon audio-based and web-based similarity measures," in *International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2007, pp. 447–454.

[17] P. Knees and M. Schedl, "A survey of music similarity and recommendation from music context data," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 10, no. 1, pp. 2:1–2:21, 2013.

[18] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, "A survey of audio-based music classification and annotation," *IEEE transactions on multimedia*, vol. 13, no. 2, pp. 303–319, 2011.

[19] J. Bergstra, "Algorithms for classifying recorded music by genre," 2006, master's thesis, Université de Montréal.

[20] M. I. Mandel and D. Ellis, "Song-level features and support vector machines for music classification," in *International Society for Music Information Retrieval Conference*, 2005, pp. 594–599.

[21] C. Xu, N. C. Maddage, X. Shao, F. Cao, and Q. Tian, "Musical genre classification using support vector machines," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5. IEEE, 2003, pp. V–429.

[22] P. Ruvolo, I. Fasel, and J. R. Movellan, "A learning approach to hierarchical feature selection and aggregation for audio classification," *Pattern Recognition Letters*, vol. 31, no. 12, pp. 1535–1542, 2010.

[23] S. Sigtia and S. Dixon, "Improved music feature learning with deep neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2014, pp. 6959–6963.

[24] P. Hamel and D. Eck, "Learning features from music audio with deep belief networks," in *International Society for Music Information Retrieval Conference*, vol. 10, 2010, pp. 339–344.

[25] L. Deng, D. Yu *et al.*, "Deep learning: Methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[26] I.-Y. Jeong and K. Lee, "Learning temporal features using a deep neural network and its application to music genre classification," in *International Society for Music Information Retrieval Conference*, 2016, pp. 434–440.

[27] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2017, pp. 2392–2396.

[28] J. J. Burred and A. Lerch, "A hierarchical approach to automatic musical genre classification," in *International Conference on Digital Audio Effects*, 2003, pp. 8–11.

[29] C. McKay and I. Fujinaga, "Automatic genre classification using large high-level musical feature sets," in *International Society for Music Information Retrieval Conference*, 2004, pp. 525–530.

[30] T. Li and M. Ogihara, "Music genre classification with taxonomy," in *International Conference on Acoustics, Speech, and Signal Processing*, vol. 5. IEEE, 2005, pp. v–197.

[31] S. Brecheisen, H.-P. Kriegel, P. Kunath, and A. Pryakhin, "Hierarchical genre classification for large music collections," in *International Conference on Multimedia and Expo*. IEEE, 2006, pp. 1385–1388.

[32] C. N. Silla and A. A. Freitas, "Novel top-down approaches for hierarchical classification and their application to automatic music genre classification," in *IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2009, pp. 3499–3504.

[33] H. B. Ariyaratne and D. Zhang, "A novel automatic hierachical approach to music genre classification," in *International Conference on Multimedia and Expo*. IEEE, 2012, pp. 564–569.

[34] W. Du, H. Lin, J. Sun, B. Yu, and H. Yang, "A new hierarchical method for music genre classification," in *International Congress on Image and Signal Processing, BioMedical Engineering and Informatics*. IEEE, 2016, pp. 1033–1037.

[35] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel, "Decision trees for hierarchical multi-label classification," *Machine Learning*, vol. 73, no. 2, p. 185, 2008.

[36] J. Wehrmann, R. Cerri, and R. Barros, "Hierarchical multi-label classification networks," in *International Conference on Machine Learning*, 2018, pp. 5075–5084.

[37] B. Z. Santos, G. T. Pereira, F. K. Nakano, and R. Cerri, "Strategies for selection of positive and negative instances in the hierarchical classification of transposable elements," in *Brazilian Conference on Intelligent Systems*. IEEE, 2018, pp. 420–425.

[38] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: a dataset for music analysis," in *International Society for Music Information Retrieval Conference*, 2017, pp. 316–323.

[39] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "LibROSA: audio and music signal analysis in python," in *Python in Science Conference*, 2015, pp. 18–25.

[40] J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques*, 3rd ed. California: Morgan Kaufmann, 2011.