Applied Soft Computing Journal 94 (2020) 106488

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Neuro-genetic programming for multigenre classification of music content

G. Campobello [c], D. Dell'Aquila [d], M. Russo [a,b,*], A. Segreto [c]

[a] Dipartimento di Fisica e Astronomia, Università degli Studi di Catania, 95125 Catania, Italy
[b] INFN-Sezione di Catania, Via Santa Sofia, 95125, Italy
[c] Dipartimento di Ingegneria, Università degli Studi di Messina, 98166, Messina, Italy
[d] Ruđer Bošković Institute, Department of Experimental Physics, Bijenička 54, Zagreb, HR-10000, Croatia

ARTICLE INFO

Article history:
Received 14 January 2020
Received in revised form 23 April 2020
Accepted 13 June 2020
Available online 20 June 2020

MSC:
00-01
99-00

Keywords:
Genetic programming
Artificial neural networks
Music genre recognition
Multi-label classifiers
Fuzzy classification

ABSTRACT

A machine learning approach based on hybridization of genetic programming and neural networks is used to derive mathematical models for music genre classification. We design three multi-label classifiers with different trade-offs between complexity and accuracy, which are able to identify the degree of belonging of music content to ten different music genres. Our approach is innovative as it entirely relies on simple analytical functions and a reduced number of features. Resulting classifiers have an extremely low computational complexity and are suitable to be easily integrated in low-cost embedded systems for real-time applications. The GTZAN dataset is used for model training and to evaluate the accuracy of the proposed classifiers. Despite of the reduced number of features used in our approach, the accuracy of our models is found to be similar to that of more complex music genre classification tools previously published in the literature.

1. Introduction

Music genre recognition (MGR) tools, i.e. algorithms capable to associate a music content to a certain or more musical genre(s), are nowadays highly demanded by services providing access to music collections and/or audio streaming. For example, most of such services, like Spotify or iTunes, widely rely on the use of music genre in order to organize and facilitate the search for music content. However, despite of its broad use, music genre is still an ill-defined concept. While the vast majority of music genre classifiers developed to date have been focused on obtaining high accuracy classification of each individual song into a single (exclusive) genre [1–6], it has been proven as several genres exhibit "fuzzy" boundaries [7]. Experimental results point out that even human agreement regarding the belonging of music content to a particular genre is less than 80% [8,9]. In addition to these facts, there are many situations in which the classification into more than one single genre is more appropriate for a given song [10]. In this framework, to contribute to improve the common agreement on the classification given by MGR tools, it is of primary importance to develop new classifiers capable to serve as multi-genres tools.

Constructing a multi-label music genre classifier is anything but an easy task. One of the major difficulties that one has to face while developing a music multi-label classifier is that the datasets commonly used to validate MGR algorithms are exclusively single-genre oriented. This is the case, for instance, of the GTZAN genre collection [11], the most used dataset for MGR [12]. GTZAN contains a collection of 1000 songs classified into ten non-overlapping music genres. In a similar dataset, each song is labeled according to the genre that *best represents* its music content. This is only in the most favorable case the prevalent genre, given that, as pointed out by the above arguments, it is not always possible to define a unique predominant genre for a given song. For the purpose of developing a multi-label classifier, GTZAN, as any other single-label datasets, should be therefore considered not fully accurate. When used to train a multi-genre classifier, such inaccuracies clearly introduce a noticeable noise in the learning set, negatively affecting the capabilities of the resulting model to correctly classify music content. Despite these aspects, we can state that the GTZAN dataset contains a significant and valuable informative content, at least in the presence of a particularly dominant genre and when the tagged genre matches with the predominant one. Recently, the GTZAN dataset has been successfully exploited to develop a multi-genre classifier called Deep Sound (DS) [13]. The DS tool, which combines Computational Neural Networks (CNN) and Long Short-Term Memory

* Corresponding author.
E-mail address: marco.russo@ct.infn.it (M. Russo).

https://doi.org/10.1016/j.asoc.2020.106488
1568-4946/

(LSTM) Networks, has achieved a discrete level of accuracy relying on a set of 518 audio features used as inputs.

In this paper, we develop a new generation approach for multi-genre classification of music content able to evaluate the degree of belonging of songs to ten music genres. We propose three different models obtained by exploiting a novel distributed software tool for non-linear regression called Brain Project (BP) [14,15], a tool capable to derive formal modeling of data using a hybridization of Genetic Programming (GP) [16] and Neural Networks (NNs), running on any 64-bit Linux machine connected to Internet via the TCP/IP protocol [17]. To train our models, we used the GTZAN dataset. The novelty of our approach consists in the construction of low complexity classifiers that are uniquely based on analytical formulas and a strongly reduced number of audio features. As an example, the simplest of our models exploits only 32 features and has an accuracy similar to that of other multi-genre classifiers published in the literature. Thanks to the deriving lower computational complexity, our models are highly feasible to be integrated in low-cost embedded systems for real-time applications.

The paper is organized as follows, Section 2 describes the problem of multi-genre classification and the details of our approach, Section 3 provides some information regarding the BP and our error measurement, in Section 4 we discuss our proposed multi-genre classifiers. Finally, in Section 5 we compare our models with other MGR tools published in the literature, while Section 6 summarizes our conclusions.

2. Multi-genre classification with single-genre oriented validation datasets: problems and approach

For simplicity, the automatic classification of music content can be divided into two distinct processes:

1. *Extraction of audio features*: this phase consists in the extraction of numerical music descriptors (also called *feature vectors*) from raw music data. Frequency and time domain parameters are often considered [18,19] but recently visual and textual features have been also exploited [20].
2. *Classification*: this is the process responsible for the association of an individual to one or more categories, based on the evaluation of its properties. In the case of MGR algorithms, individuals are audio signals, categories are music genres and properties are audio features [7].

More formally, given X the domain of feature vectors and Y the one of labels characterizing music genres (e.g. Rock, Pop, etc.), a multi-label classifier can be defined according to two functions: $f : X \times Y \rightarrow \Re$ and $h : X \rightarrow 2^Y$. The function $f(x, y)$ can be regarded as the confidence of $y \in Y$ being the proper label for $\mathbf{x} = \{x_i\}$. Naturally, f should return a larger output as the degree of belonging of the input features $\mathbf{x}$ to y is larger. The function h is instead used to predict the correct set of labels that are relevant for $\mathbf{x}$. This is usually expressed in terms of the function f and an additional function $thr : X \rightarrow \Re$, typically referred as *thresholding function*, in the following way: $h(\mathbf{x}) = \{y | f(\mathbf{x}, y) > thr(y), y \in Y\}$. The main problem in the case of multi-label classification consists in the determination of the function f, and, eventually, the function thr [21]. The latter is typically a rather difficult task. For instance, in MGR applications, thresholds might not exist at all and the thresholding process, i.e. the *defuzzification*, could destroy or hide information. For this reason, in this work, we concentrate our attention mainly on the function f, while the search for optimal thr functions can be customized according to the desired application.

The proposed classification system is based on a set of analytical functions $\{f_j : X \rightarrow \Re | 1 \leq j \leq N_l\}$, which, given an input feature vector $\mathbf{x}^{(i)}$ representing the ith song, provide the degree of belonging of the song to each of N_l genres, independently. We consider $N_l = 10$ music genres: blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae and rock. Hereafter, we indicate such degrees of belonging with a vector $\mathbf{c}^{(i)} = [c_1^{(i)}, \ldots, c_{10}^{(i)}]$, where $c_j^{(i)} = f_j(x^{(i)})$. It is important to stress that the independent evaluation of the belonging to each of the N_l genres is a crucial aspect of our approach. Let us consider for example the case of the song *Barcelona*, recorded by the operistic soprano Montserrat Caballé and Freddie Mercury, front-man of the popular rock band Queen. This song is a classical crossover where opera, pop and rock genres are merged together [22]. It is clear as a proper classification of this song should result in a high degree of belonging to at least three individual genres simultaneously. Additionally, one can also state that an almost full degree of belonging of this song with the rock genre should not exclude an equally large degree of compatibility with, for instance, the classical genre.

The derivation of MGR functions relies on the use of a database of music content for the so-called *model training*, i.e. a learning phase where the algorithm extracts information from a series of given examples. The classification approach used in this paper falls in the field of supervised learning. Supervised approaches consist in the training of a model against a number of input feature vectors for which the output is known a priory. This is typically the case for MGR tools, where one can exploit databases of previously labeled music content to train the model. Model training is probably the major challenge in the development of multi-genre classifiers because music databases for MGR validation are labeled according to a single-genre philosophy. The problem can be simplified in the following way. In a learning database, each input feature vector, i.e. the ith music content to classify, is associated to a certain genre with a boolean vector $\mathbf{c}^{(i)}$. In other words, for each training input, one has 10 available information, one for each music genre considered in the analysis: the belonging to a genre and the non-belonging to the rest 9 genres. This carries a significant informative content as soon as the song represented by the training input is characterized by a particular predominant genre. For those cases in which, as discussed in the above example, two or more labels are appropriate, the corresponding information is imprecise for all the coefficients $c_j^{(i)}$ considered as zero and representing significant labels for the given input.

From the above arguments, we can state that the information contained in single-genre oriented databases is noisy with regard to the validation of multi-label classifiers. The problem of the derivation of a multi-label classifier corresponds therefore to that of a supervised regression approach in the presence of a noisy dataset. In this framework, overfitting can become an extremely important problem [23]. This statement is better clarified by the example described in Fig. 1, where we schematically represent the modeling of a dataset in the presence of noise. Let us assume that the data to model is generated according to the *exact model* represented by the solid line in figure and by adding an arbitrary noise (open symbols). The task of a modeler will be the one of deriving a model that best approximates the exact one, which is unknown to the modeler. To this end, we have considered two different possible analytical models characterized by different complexities: a high-complexity model and a low-complexity one. The first is obtained by using a high-order polynomial (40th-order in the example) while the latter is represented by a low-order polynomial (namely 2nd-order). As one can semi-quantitatively observe, it is clear that the higher-complexity model better approximates the samples (dashed black line), i.e. minimizes the discrepancies with the dataset, however, the lower-complexity one (dot-dashed red line) is significantly

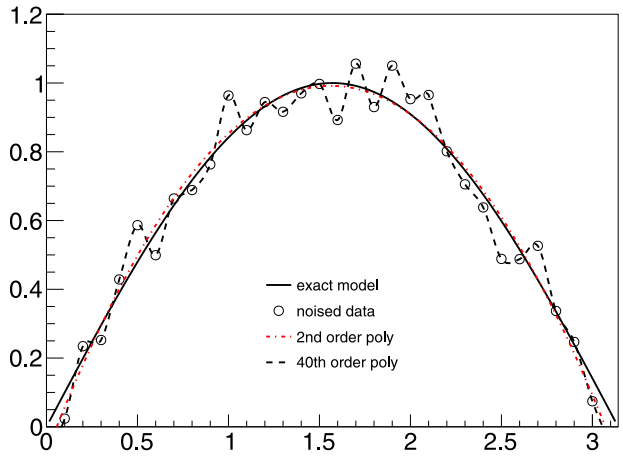

Fig. 1. An example of overfitting. Open circles are the data samples to model, generated according to the *exact model* represented by the solid line and by adding a stochastic noise. The resulting noised dataset is modeled by using two possible analytical models: a 2nd order polynomial function (dash-dotted red line) and a 40th order polynomial function (dashed line). The higher degree of complexity of the 40th order polynomial function results in the overfitting of data, while the reduced complexity model is capable to extract the underlying information being less affected by noise.

closer to the exact model. In other words, the reduced complexity model was able to filter-out the noise and to extract the useful underlying information from the training dataset even in the presence of fluctuations. On the contrary, the higher-order polynomial model has significantly better performances when applied to the training dataset, at the cost of an obvious lack of generality and a consequent undesired behavior when applied to new data. More quantitatively, the overfitting problem is closely related to the ratio of the quantitative of memory available to the modeler and the number of data samples used to train the model. Naturally, for a fixed complexity of the model, overfitting becomes less severe as the size of the dataset increases.

The database used for the validation of our models is the GTZAN music collection. It is important to stress that, because of the relatively limited size of this dataset (only 1000 songs are provided) and because of its intrinsic single-genre content, in the present application it is crucial to opportunely limit the complexity of the models in order to avoid overfitting and to extract information useful to multi-label classification purposes. Our reduced complexity models are derived by exploiting a series of BP features.

3. The brain project

GP falls in the field of Evolutionary Computing (EC) [24]. Within this field, a particular line of research is the one of Symbolic Regression (SR). SR is the process during which some data is fitted by means of a suitable analytical formula. With this respect, many novel approaches have shown that the evolutionary core of GP is capable to solve extremely complex problems. Additionally, some GP implementations are also capable to deal with the so-called feature selection, i.e. the dimensionality reduction of input variables. This aspect is particularly important when the cardinality of data to fit is significant, as in the case of MGR (see Section 4).

The Brain Project is a hybridization of GP and NNs. These two aspects, hereafter indicated as BP-EC and BP-NN, cooperate with the aim of deriving the proper analytical model $g(\mathbf{x})$ to fit a set of data $\{(\mathbf{x}_i, \mathbf{y}_i)|\ 1 \leq i \leq N_p\}$. The BP-EC approach is part of a line of GP research that foresees the evolution of tree structures

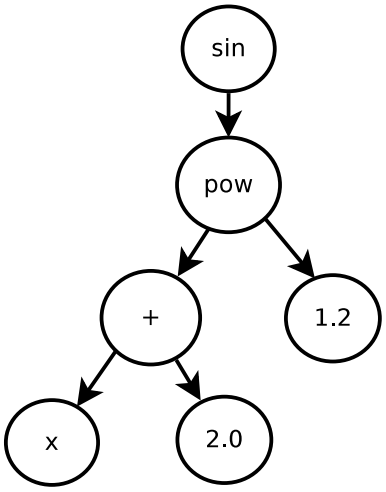

Fig. 2. Tree representation of the expression $\sin\left((x+2.0)^{1.2}\right)$.

representing formal mathematical expressions. To this end, four different steps are iterated:

1. Generate an initial population of random trees representing possible solutions to the SR problem being studied;
2. Evaluate each possible solution and assign it a fitness value f_{fit};
3. Create a new population of trees starting from the previous population using some evolutionary operators, e.g. copy, crossover, mutation, selection and heuristic operators;
4. Return to step two until the termination criterion is met.

In the end, the best individual, i.e. the tree with the maximum value of f_{fit}, is designated as the output result.

The global search operated by BP-EC is extremely powerful to identify a solution close to a global minimum of the fitness function, but its convergence speed to the minimum itself is usually low. BP-NN is therefore used as a hill-climbing operator. It consists in a fast local search for the minimum of the fitness by suitably varying the constants. Because this is a particularly time consuming task, only when one individual is particularly promising (i.e. close to a local minimum of the fitness) BP-NN is invoked. In BP-NN, each expression is firstly transformed into a multi-layer feed-forward neural network by replacing operators in the original expression with specialized neurons, while all constants are considered as weights of the NN. The mathematical expression of the error gradient in the weight space is then calculated and an enhanced steepest descending technique is used to update the weights. Separate learning rates for each weight are dynamically modified during the training.

Currently, BP consists of some tens of thousands of C code lines. It is composed of a set of highly optimized, object-oriented modules for vector processing. Its internal data structures were designed considering CPUs as Single-Instruction-Multiple-Data processors. One of the most salient features of the BP is its distributed implementation. Nevertheless the only requirement is the TCP/IP protocol, i.e. neither Message Passing Interface (MPI) nor Parallel Virtual Machine (PVM) nor any other overhead is needed [25]. All that is required is simply a set of interconnected commodity off-the-shelf computers with a 64-bit Linux operating system.

BP organizes its data structures, representing formal expressions, in tree-like structures. The complexity of a mathematical expression is computed by simply counting the number of nodes. An example is shown in Fig. 2 where the tree representation of the expression $\sin\left((x+2.0)^{1.2}\right)$ is reported. This expression, for instance, comprises 6 nodes.

3.1. The error measurement and fitness function

The fitness function f_{fit} can be expressed in terms of two factors, f_e and f_n, related, respectively, to the approximation error and the complexity of the mathematical expression, i.e. the corresponding number of nodes. The desired maximum number of nodes, $n_{\max}$, can be specified in a configuration file of the BP, so that the factor f_n can be expressed as

$$f_n = \frac{n_{\max} - n}{n_{\max} - 1} \tag{1}$$

Note that values of f_n are lower than or equal to 1 and are negative when $n > n_{\max}$. It is worth mentioning that other complexity metrics can be used in BP. For instance, it is possible to assign different costs to different operators.

The factor f_e is defined as

$$f_e = \frac{e_{\max} - e}{e_{\max}} \tag{2}$$

where $e_{\max}$ is the maximum error specified in the configuration file while the error e is evaluated according to [26] as

$$e = 100 \cdot \sqrt{\sum_{i=1}^{N_p} \sum_{j=1}^{L} \frac{w_{ij}}{S_w} \left(\frac{y_{ij}^c - y_{ij}^d}{\sigma_j} \right)^2} \tag{3}$$

having y_{ij}^d the desired output, i.e. $y_{ij}^d = 1$ if the music content i belongs to the genre j ($y_{ij}^d = 0$, otherwise), $y_{ij}^c = g_j(\mathbf{x}_i)$ the jth calculated output corresponding to the ith pattern, L the number of outputs/genres, N_p the number of learning patterns, w_{ij} a weight factor that can be assigned to the jth output of the ith pattern, $S_w = \sum_{i=1}^{N_p} \sum_{j=1}^{L} w_{ij}$ the sum of all weights, $\sigma_j = \sqrt{\frac{\sum_{i=1}^{N_p} w_{ij}(y_{ij}^d - \mu_j)^2}{\sum_{i=1}^{N_p} w_{ij}}}$ the jth output weighted standard deviation, and $\mu_j = \frac{\sum_{i=1}^{N_p} w_{ij} y_{ij}^d}{\sum_{i=1}^{N_p} w_{ij}}$ the jth output weighted mean. Note that weights make it possible that some data points and/or outputs contribute more than others, i.e. we can use higher weights to state that related outputs are more important than others. This feature can be useful in the case of datasets with an unbalanced number of songs or to emphasize the importance of true positive results.

The following expression for the total fit function f_{fit} is used by BP:

$$f_{\text{fit}} = 100 \cdot \frac{f_e u(f_e) + \alpha f_n u(f_n)}{1 + \alpha} \cdot \cdot \exp(f_e(1 - u(f_e)) + f_n(1 - u(f_n))) \tag{4}$$

where $u(x)$ is the Heaviside step function and α is a coefficient usually smaller than 1. The latter is an heuristic factor used to control the relative importance of complexity and accuracy in the formation of f_{fit}. In particular, a large α value would result in the derivation of particularly simple but inaccurate solutions, while a small α value would most likely result in the convergence of the algorithm towards more accurate but more computationally complex solutions. For the purpose of this work, we used $\alpha = 0.01$. The Heaviside step function is introduced in the above expression to smooth the effects of solutions with a complexity greater than $n_{\max}$ or with an error greater than $e_{\max}$. The exponential term $\exp(f_e(1 - u(f_e)) + f_n(1 - u(f_n)))$ is finally used to avoid discontinuities in the derivatives of f_{fit} with respect to f_e and f_n, respectively at $e = e_{\max}$ and $n = n_{\max}$.

In the configuration file it is also possible to set a target error, e_{target}. In this case a target error is given, BP tries to obtain the simplest solution with the desired error. This is obtained by multiplying the above expression of f_{fit} by the factor $\exp(\frac{e_{\text{target}} - e}{e_{\text{target}}})$. As a consequence f_{fit} further decreases for all solutions such that $e > e_{\text{target}}$.

Finally, the vast majority of parameters used in BP are self-adaptive, as carefully described in the dedicated analysis of Ref. [15].

4. Our new multi-genre classifier

This section describes the derivation of three distinct analytical multi-label MGR models obtained by exploiting the unique features of the BP. The database used to validate the models, i.e. the GTZAN music collection, is presented. We discuss the audio features commonly used in MGR approaches and the proposed feature reduction obtained through our model training procedure. Finally, we provide and discuss the derived mathematical formulas for all the proposed classifiers.

4.1. The GTZAN dataset

Data used to derive and validate our models are extracted from the GTZAN music collection [11]. This dataset, created by Tzanetakis and Cook, is the first *benchmark* dataset made publicly available for MGR. To date, it is the most used dataset for MGR. The GTZAN music collection provides 1000 individual audio tracks each 30 seconds long. Such audio content is subdivided into 10 non-overlapping genres: Blues, Classical, Country, Disco, Hip-Hop, Jazz, Metal, Pop, Reggae, Rock, each containing an equal number of tracks. To account for possible mislabeled content, as evidenced for example in Ref. [27], we developed a preliminary classification model with the aim to identify and remove some possible faults in the GTZAN dataset. Our filtering procedure is described in Section 4.3.

4.2. Audio features

Features have been extracted from the raw data representing the audio files using the Python library *librosa* [28]. In particular, we used the Bag-Of-Features (BOF) approach to model each audio signal considering the long-term statistical distribution of its short-time spectral features. This process can be described as follows. In a first step, raw data have been divided into overlapped frames of 2048 samples each applying a Hanning window of 512 samples. For each of the considered frames, the short term features summarized in Table 1 have been evaluated. A description of such quantities is provided, for example, in [29]. An overall number of 74 parameters has been evaluated for each frame. The statistical distributions of extracted short term features have been analyzed. In particular, with the aim of reducing the number of parameters used for classification, we evaluated for each parameter only 7 statistical values, i.e. Average (A), Standard deviation (S), skeWness (W) and Kurtosis (K), together with Lowest (L), Median (M) and Greatest (G) value. At the end of this process, each audio file is thus represented with a feature vector $\mathbf{x}$ composed by $74 \times 7 = 518$ real values.

Hereafter we refer to the these values using the notation $acronymX_n$ where *acronym* represents one of the acronyms reported in Table 1, while $X \in \{A, S, W, K, L, M, G\}$ is a symbol indicating the corresponding statistical value and n is an index used to specify the short-term feature parameter. For instance, with $tonzK_6$ we indicate the Kurtosis of the 6-th tonal centroid.

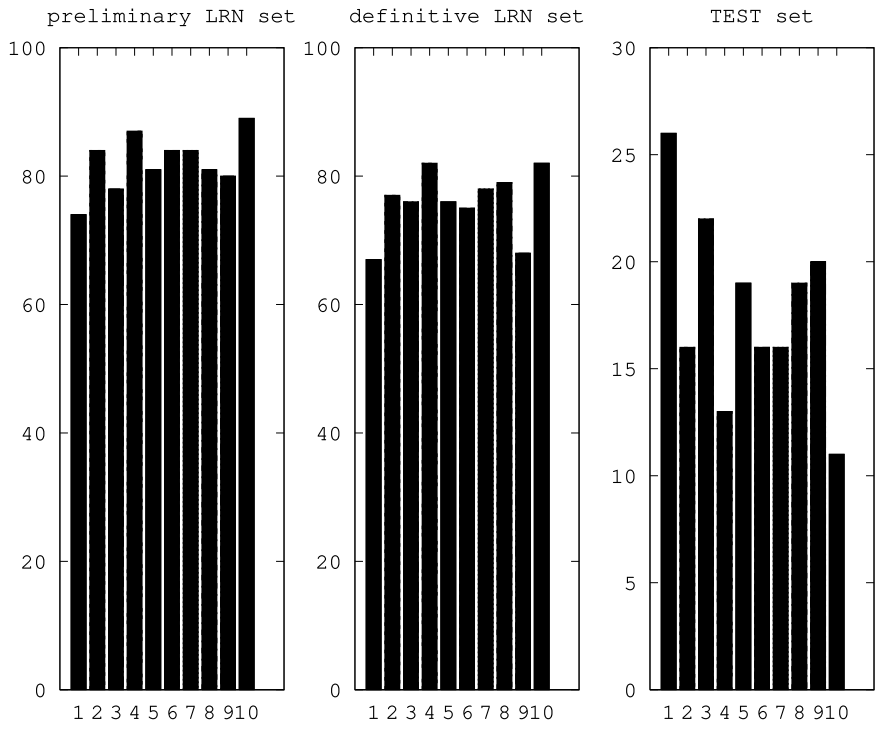

Fig. 3. Music genre distribution for the preliminary learning set (left panel), the filtered learning set (middle panel) and the testing set (right panel). The x-axis index represents the given genre in the following numerical order: blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, rock. Observed fluctuations in the number of excerpts for each genre reflect the statistics of our random extraction procedure.

Table 1
List of short-term spectral features evaluated from the extracted frames. The reader can refer for example to Ref. [29] for a short description of these quantities. The distributions of such features for all the selected frames are analyzed in terms of 7 selected statistical values to serve as audio features for our models. See text for details.

Feature name	Acronym	# of parameters
Zero crossing rate	zcr	1
Constant Q-Chromagram	chcqt	12
Chroma energy normalized	chcens	12
Tonal centroid	tonz	6
Chromagram	chstft	12
Spectral centroid	scent	1
Spectral bandwidth	spbw	1
Spectral contrast	scont	7
Spectral rolloff	sroll	1
Spectral flatness	sflat	1
Mel-frequency cepstral coefficients	mfcc	20

4.3. Model training and feature selection

As the first step of the derivation of our models, we randomly assigned each of the excerpts provided by GTZAN to either learning (LRN) or testing (TST) datasets. The LRN set is used to train our models while the TST one is used to evaluate the quality of model learning. To obtain an unbiased distribution of music content, with regard to each individual genre, we proceeded as follows: for each of the excerpts we extract a random number r uniformly distributed in the interval $[0, 1]$; we include the excerpt in the LRN set if $r < 0.8$, otherwise we assign it to the TST set. At the end of the selection procedure, we obtain a preliminary learning set of 822 excerpts and a testing set given by the remaining 178. The genre distribution of the preliminary LRN set and the TST set obtained with our procedure is shown, respectively, in the left and right panels of Fig. 3, where an index from 1 to 10 is used to refer to each of the music genres considered in this analysis with the same numerical order as mentioned in Section 2. The small differences between the number of audio tracks selected for each genre are the result of the statistical nature of our approach.

The quality of the information contained in the GTZAN dataset is affected by a number of factors, including multi-genre songs classified as single-genre (see Section 2) and incorrectly labeled songs [27]. Reducing any possible source of noisy information is mandatory in order to enhance the generality of the resulting classifiers. To this end, we removed from the preliminary LRN set a number of possible mislabeled tracks identified by adopting the approach described below.

The LNR set is first used to train a preliminary classifier. After a quick learning phase, we evaluated the content of the LNR dataset with the preliminary classifier obtaining an output vector $\mathbf{c}$ for each of the excerpts in the dataset. Let $g^{(i)}$ the index corresponding to the ground truth genre specified by GTZAN for the ith song and $\mathbf{c}^{(i)}$ the corresponding output evaluated by our preliminary classifier. For a correctly tagged song with a single genre and in the presence of a perfect classifier, it holds trivially $\max\{c_j^{(i)}\} = c_{g^{(i)}}^{(i)}$. We assumed that a song is misclassified if $\max\{c_j^{(i)}\} > 10 c_{g^{(i)}}^{(i)}$.

By applying the procedure discussed above, we identified 62 possible mislabeled songs that were removed from the preliminary LRN dataset. The resulting filtered dataset is thus composed by 760 excerpts, with the genre distribution shown in the central panel of Fig. 3. Also in this case, songs are equally distributed between genres within the statistical uncertainties. In the following, we adopt the filtered LRN set as the learning set for our models.

Our classification models are based on a set of 10 distinct classifiers, each returning an output in the range $[0, 1]$. Each classifier is trained separately by using all the songs in the LRN dataset. In a similar procedure, a classifier is trained with a number of examples of which 10% have a full degree of belonging to the genre itself ($c_j^{(i)} = 1$) and the rest 90% have a vanishing degree of belonging (i.e. $c_j^{(i)} = 0$). As a consequence of this asymmetrical number of examples, resulting classifiers will be

Table 2
Summary of all the derived classification models. n_{LRN} and n_{TST} are the number of patterns used respectively in the learning set and testing set, e is the BP error evaluated according to Eq. (3), n_{nodes} is the overall number of nodes used for the BP tree structures representing the model formulas, $n_{features}$ is the corresponding number of selected features.

Model	n_{LRN}	n_{TST}	e Eq. (3)	n_{nodes}	$n_{features}$
Preliminary	822	178	75.6%	235	64
CM1	760	178	65.7%	250	79
CM2	760	178	70.0%	179	58
CM3	760	178	75.0%	89	38

Table 3
List of the 38 features exploited by CM3 according to the notation introduced in the Section 4.2.

$chcensA_3$	$chcensL_{10}$	$chcensL_2$	$chcensM_9$	$chstftM_{10}$
$chstftM_9$	$chstftW_5$	$mfccA_{11}$	$mfccA_{17}$	$mfccA_4$
$mfccA_6$	$mfccA_9$	$mfccG_{11}$	$mfccG_{20}$	$mfccK_1$
$mfccM_{15}$	$mfccS_{10}$	$mfccS_{12}$	$mfccS_5$	$mfccS_9$
$mfccW_{16}$	$mfccW_7$	$scentS_1$	$scontA_5$	$scontG_1$
$scontK_4$	$scontK_5$	$scontL_1$	$scontL_5$	$scontS_5$
$scontW_4$	$sflatA_1$	$sflatG_1$	$srollL_1$	$srollS_1$
$tonzS_3$	$tonzS_4$	$zcrW_1$		

maximally trained to recognize the non-belonging of music content to genres. In other words, we expect to obtain *conservative* classifiers, i.e. classifiers that return a meaningful classification to a given genre only when a particularly significant correlation with the input features is present.

In the present work, we derive 3 distinct classifiers:

- The Classification Model 1 (CM1) has been obtained without constraints on the resulting complexity and is therefore the one with largest complexity and greatest accuracy. The corresponding error e, evaluated according to Eq. (3), is equal to 65.7%.
- The Classification Model 2 (CM2) has been obtained by imposing a target error equal to 70%, i.e. an increase of about 5% in the error level with respect to the one achieved by CM1.
- The Classification Model 3 (CM3) has been obtained considering a target error equal to 75%. This is the simplest of the models we developed.

In order to ensure the output of each of the classifiers fulfilling the condition $f_j(\mathbf{x}) \in [0, 1]$ for all possible input vectors $\mathbf{x}$, we replaced f_j with the corresponding constrained functions $g_j(\mathbf{x}) = max(min(f_j(\mathbf{x}), 1), 0)$.

Finally, the entire simulation procedure adopted for the derivation of CM1-3 is schematically described in Fig. 4.

We summarize in Table 2 our derived models. Differently from the preliminary model, CM1-3 are trained by using the reduced LRN set ($n_{LRN} = 760$). It is worth noting that the error e reported in the table does not coincide with the accuracy of the model. The latter will be defined in Section 5 and used to compare our models with other existing models published in the literature. The computational complexity of the derived models is indicated through the number of nodes n_{nodes} used by BP to describe their analytical formulas. As it is possible to observe by inspecting the results reported in Table 2, the lower is the target error used to derive the model, the lower is the corresponding computational complexity. As an example, the most accurate model, CM1, comprises 250 nodes while only 89 nodes are required to implement CM3. Concerning the number of features required to classify the input content, a highly significant reduction is done thanks to the feature selection operated by BP. For example, only 79 features, about 20% of the 518 features usually adopted in the literature, are used by CM1. The information contained in the remaining 439 features is found to be not particularly significant to the classification process. Interestingly, when the complexity of the model is reduced, additional features are found to be less significant. The simplest of our models, CM3, exploits only 38 features, less than 8% of the original number of features used in the learning process. By using such a reduced number of features, the corresponding complexity amounts at 32% of that of CM1, with a worsening of the error of less than 10%. As it will be discussed in Section 5, the accuracy of CM3 in music content classification is only slightly worse than the one of CM1. A complete list of the features foreseen by CM3 is reported in Table 3 according to the notation introduced in Section 4.2.

4.4. Performance of the models

The analytical formulations of CM1-3 are given, respectively, in Figs. A.8, A.9 and A.10 of Appendix. They can be easily used, given a set of input features, to calculate the degree of belonging of an audio track described by a set of features to each of the 10 music genres considered in this analysis. To have a first, semi-quantitative, idea about the capabilities of our classifiers in the task of MGR, we apply the proposed classifiers to the entire GTZAN dataset. In Fig. 5, we report the results of our analysis. Each subfigure provides an image of 1000 × 10 wide pixels (horizontal segments) where each row is associated to an individual excerpt and each column is related to a certain music genre. The generic pixel (i, j) is black if the genre j is a good classification for the song i and white otherwise. Note that, in order to improve the readability of the figure, on the x-axis we reported directly the name of genres instead of the corresponding number $j \in [1, 10]$. The first subfigure in the left panel of Fig. 5 represents the GTZAN dataset as analyzed by an ideal single-genre classifier. Each group of 100 consecutive songs belongs exclusively to a given genre. The resulting picture contains sharp bands, each corresponding to one of the music genres included in the database. For example, all songs in the range 300 to 399 are labeled as blues, while all songs in the range 700 to 799 are tagged as classical and so on. The presence of well-defined sharp bands is because the belonging is here considered exclusive to one single genre. For all the other subfigures, related to our classification models, the generic pixel (i, j) is drawn in black when $c_j^{(i)} \geq 0.5$. In this case, differently from the case of the ground truth of the GTZAN dataset, more than one genre can be simultaneously present for the same y-coordinate, i.e. a given song can simultaneously belong to more than a single genre. Several important indications can be deduced by carefully inspecting the subfigures relative to CM1-3. The first important fact to note is that the number of songs for which at least one of the classifiers returns a value larger than or equal to 0.5 (i.e. the number of rows in the plot containing at least a horizontal segment) decreases while moving from left to right through the subfigures. This fact is not surprising as the model tends to be more conservative while reducing its analytical complexity. As an example, CM3 identifies only 458 songs within the GTZAN dataset that have a degree of belonging greater than or equal to 0.5 to at least one music genre. By looking at subfigure relative to CM1, groups of pixels corresponding to each individual genre are very well visible, testifying that our larger complexity model preserves in a quite satisfactory way the music genre partitioning of the original GTZAN dataset. Poorer performances if compared to the other genres are however registered for country, disco and rock. The latter is particularly troubling. If one looks at the corresponding picture for CM2 and CM3, it is clear that the performances in identifying rock content are dramatically worse while simplifying the model. At the limit, when considering CM3, one finds no rock songs at all resulting in an output greater than 0.5 for one of the genres. Such a degradation of the rock identification capabilities for simpler models suggests that the set

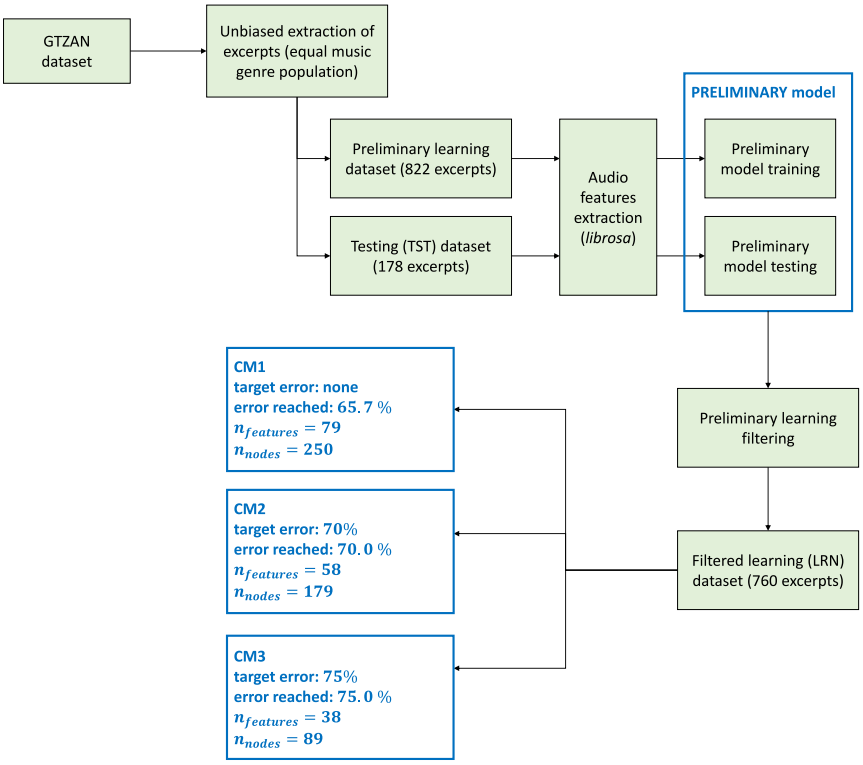

Fig. 4. A schematic representation of the simulation process performed to construct CM1-3 classification models. The BP neuro-genetic tool is used for model training.

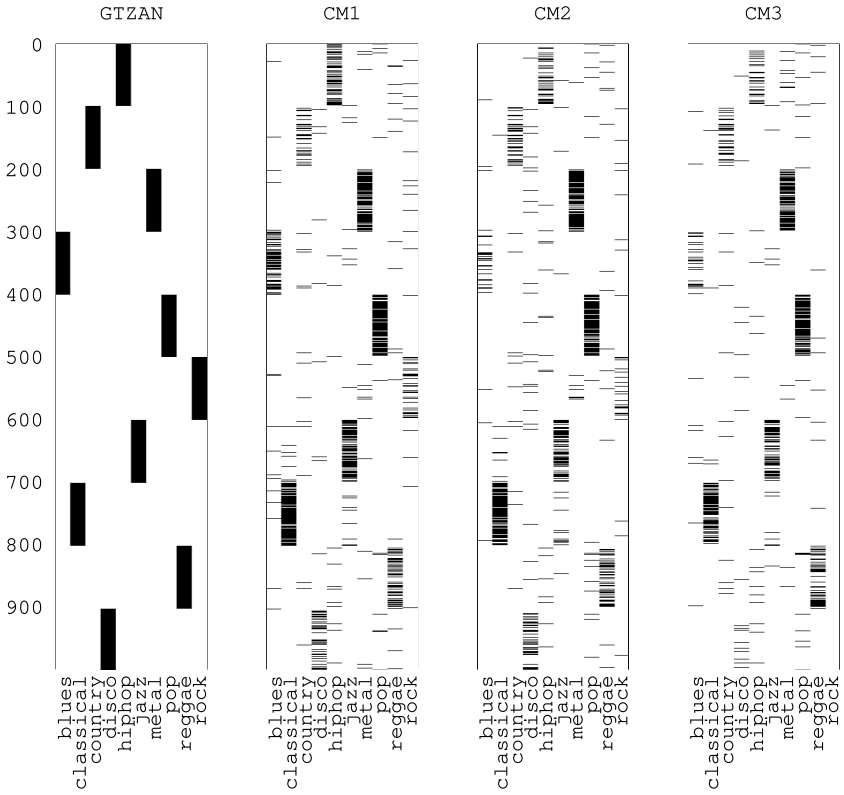

Fig. 5. Comparative plots for CM1-3 obtained by using the content of the GTZAN dataset. The *y*-axis is the index representing the *i*th content of the GTZAN dataset, while the *x*-axis reports the music genres considered in this analysis. The left panel shows the result of the application of an ideal classifier to the GTZAN dataset, an individual horizontal segment for each song describes its belonging to exclusively one music genre. All other panels are relative to CM1-3. In this case a horizontal segment is drawn only when the classifiers returns a value $c_j^{(i)} \geq 0.5$, and more than one horizontal segments on the same row is possible.

of features described in Section 4.2 contains a poor correlation to rock content. This is also confirmed by the corresponding analytical formula, f_{10} of Fig. A.10, containing only 3 nodes and exploiting exclusively one single feature among those in Table 1. Furthermore, the fact that GTZAN contains several mislabeled songs in the rock genre [27] and, at the same time, that many of the songs labeled according to other genres have a significant rock content might negatively affect the capabilities of the resulting rock classifier. Another interesting fact is that loci relative to classical, jazz, metal and pop genres in Fig. 5 are particularly marked for all our proposed classifiers. The case of CM3 is particularly remarkable. Derived formulas for these genres, as reported in Fig. A.10, have an extremely reduced complexity. Their number of nodes ranges from 6, with only 2 exploited features, for the classical classifier to 18, with 6 features, for the jazz classifier. Results are particularly interesting especially for classical, metal and pop genres, for which an extremely high density of black pixels is observed in Fig. 5, in agreement with the ground truth of GTZAN.

A more quantitative analysis of the performances of CM3 in evaluating the content of the GTZAN dataset can be done by inspecting Fig. 6. In this figure, we report 10 individual panels, each for a given music genre classified in GTZAN. In each of the panels we show a graphical representation of the coefficients $c_j^{(i)}$ obtained via CM3 for all songs (numbered from 00 to 99 in the x-axis) whose ground truth corresponds to the given genre. Dots in the figure have radii proportional to the output coefficients $c_j^{(i)}$ only when $c_j^{(i)} \in [0.5, 1]$, while no dot is drawn for $c_j^{(i)} < 0.5$. The figure confirms what, semi-quantitatively, was visible in Fig. 5. Excellent performances are obtained for the classical, metal and pop classifiers. Even by considering a quite large threshold value, the classical classifier returns a meaningful degree of belonging for 70 songs; for 63 of those, all classified as classical by the GTZAN collection, the classical degree of belonging according to CM3 results the dominant one. Concerning the metal classifier, we obtain 83 songs with a meaningful metal output; in this case, 76 songs are dominantly associated to the metal genre by CM3. Interestingly, 73 of these songs are identified as metal by GTZAN, while the remaining 3 are labeled as rock, testifying the good capabilities of the CM3 metal classifier. The CM3 pop classifier returns a value $c_j^{(i)} \in [0.5, 1]$ for 90 songs belonging to the GTZAN dataset; for all these songs, the pop genre results the dominant one according to CM3. Among these songs, 76 are labeled as pop in GTZAN, while the large majority of the rest belong to music genres with a reasonable degree of superposition with the pop genre: 3 are labeled as reggae by GTZAN, 4 are labeled as hip-hop and 5 are labeled as disco.

Finally, it is interesting to analyze the behavior of CM3 if different threshold values are used. Table 4 reports the number of songs of the GTZAN dataset with a meaningful CM3 classification for several threshold values ranging from 0.3 to 0.6. It is clear how the number of classified songs rapidly increases as a lower threshold is used; for example, only 380 songs are classified if a threshold of 0.6 is used, while we obtain 704 classified songs for a threshold value of 0.3. As a result of the lower level of conservativity, the number of true positives, i.e. the number of songs whose prevalent genre identified by CM3 coincide with the ground truth of GTZAN, slightly decreases when lower thresholds are used. However, such values are satisfactory for all considered threshold values, ranging from 90% to 80%, respectively for the maximum and minimum thresholds considered in our analysis. The number of meaningful genres is also affected by the size of the threshold. The last column of Table 4 reports the maximum number of meaningful genres for each of the corresponding threshold values, such numbers do not exceed 4 for the minimum threshold considered in this study.

Table 4
Number of classified songs, percentage of true positives and maximum number of genres identified by CM3 for different threshold values.

Threshold	# of classified songs	% of true positives	Max number of genres
0.6	380	90%	2
0.5	458	87%	3
0.4	566	84%	3
0.3	704	80%	4

Table 5
Accuracy of state-of-the-art MGR models and the proposed CMs when used as single-label classifiers, calculated according to Eq. (5). Input data are extracted from the GTZAN database.

Model	Method	Type of classification	Accuracy (Acc_{SL})	$n_{features}$
[5]	SAHS	SL	97%	265
[6]	EMD	SL	97%	616
[30]	NMPCA	SL	84%	768
[31]	EDSP	SL	85%	768
[32]	LSDR	SL	91%	768
[33]	TPNTF	SL	94%	7680
[13](DS)	NNs-SVM	ML	74%	518
CM1	EC-NNs	ML	77%	79
CM2	EC-NNs	ML	75%	58
CM3	EC-NNs	ML	70%	38

5. Comparisons with other approaches

Let us summarize the main features of our novel approach. Our classifiers are innovative as they are uniquely based on fully analytical formulas (Figs. A.8, A.9, A.10). Differently from the majority of the MGR tools previously developed, they are capable to simultaneously evaluate the degree of belonging of a given music content to several music genres with a *fuzzy-like* logic, accounting for the existence of fuzzy boundaries between music genres. In addition, our classifiers rely on a strongly reduced number of features.

To fully validate our newly developed models, it is mandatory to benchmark the accuracy of our predictions against those of well-established state-of-the-art models for music content classification. To this end, we adopt the metric of the mean accuracy (Acc), widely suggested in the literature, to compare the performances of our models with that of other MGR single- (SL) and multi-label (ML) classifiers available in the literature.

We define the mean accuracy of a classification model in the following way:

$$Acc = \frac{1}{L}\sum_{j=1}^{L}\frac{TP_j}{TOT_j} \tag{5}$$

where TP_j is the number of true positives obtained for the jth genre/class, TOT_j is the number of patterns in the jth class and L is the overall number of classes, i.e. the number of independent classifiers. The definition of mean accuracy given by Eq. (5) assumes implicitly that each input is classified into one, and only one, of L non-overlapping classes. In order to use this metric, it is therefore mandatory to force our multi-label classification models to work as single-label classifiers. It is important to stress that such a condition is required to use the simple formalism given by Eq. (5) for numerical comparisons between classifiers, but is clearly a highly unfavorable condition to our models.

Table 5 reports a full numerical comparison between the performances of our models and the performances of single- [5,6,30–33] and multi-label [13] classifiers previously published in the literature. We used the content of the GTZAN dataset to produce the comparative results. For the calculation of Acc in the case of multi-genre classification models, we consider the

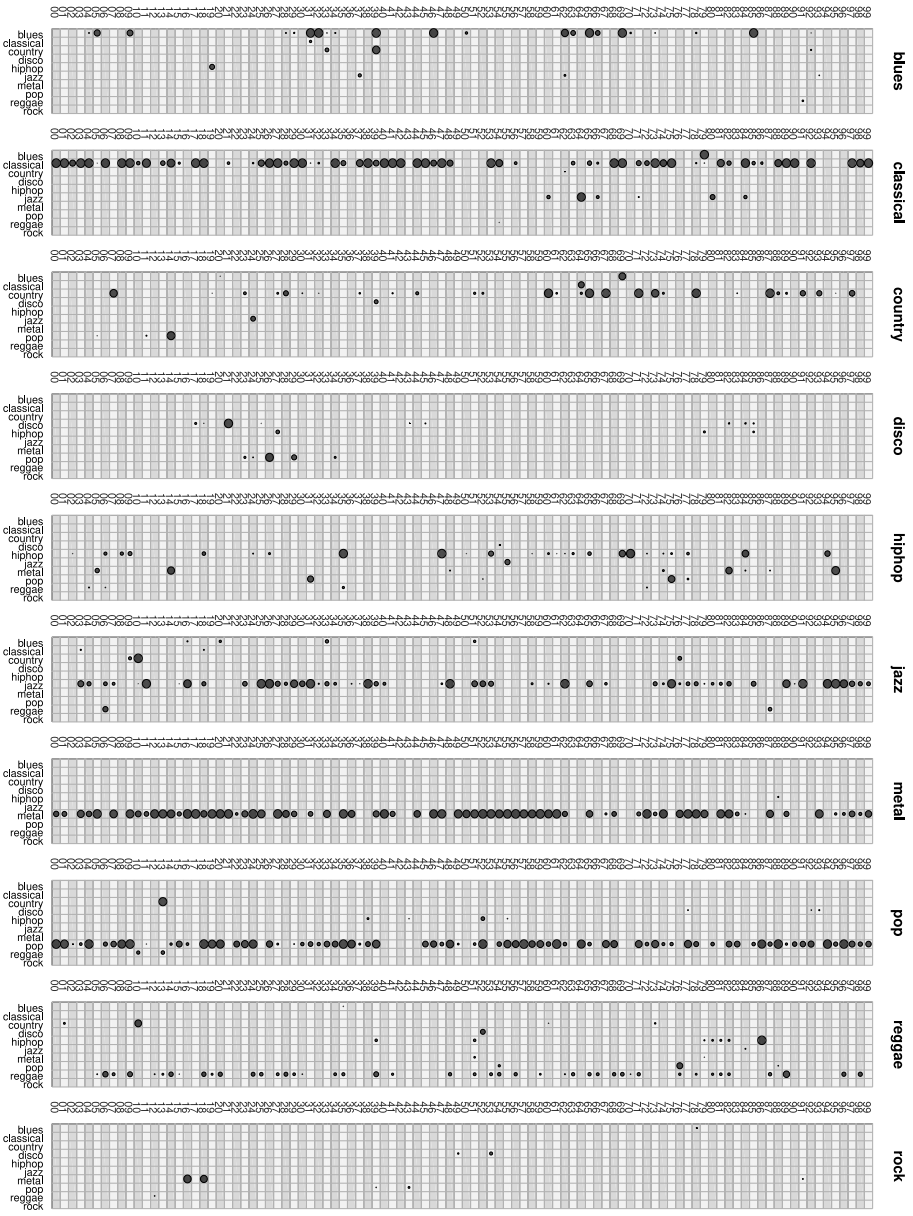

Fig. 6. Graphical representation of the coefficients $c_j^{(i)}$ obtained by CM3 for the whole GTZAN dataset. The size of the dots reflect the magnitude of the corresponding coefficient. No dot is drawn for $c_j^{(i)} < 0.5$. Each individual panel is relative to the content belonging to a certain music genre labeled by GTZAN.

ith music content as belonging to the genre g if $\max\{c_j^{(i)}\} = c_g^{(i)}$. A first important point is to compare the capabilities of our models used as single-genre classifiers with those of state-of-the-art single-genre classifiers. This is a fundamental step to validate the capabilities of our models in preserving the information on the dominant genre given by GTZAN. Accuracy values of CM1-3 are, respectively, 77%, 75% and 70%. These values are lower than those obtained with SL models, which range from 84% for [30] (which exploits non-negative multi-linear principal component analysis, NMPCA) to 97% for the most accurate ones [5,6]. The latter are based, respectively, on a self-adaptive harmony search (SAHS) algorithm and empirical mode decomposition (EMD). This result is quite satisfactory if we consider that our models are not conceived as high-precision SL models. Furthermore, the good accuracy obtained by our models when used as SL classifiers indicates that the original information given by the GTZAN dataset is reasonably preserved. Concerning the number of features $n_{features}$ exploited by the models (last column on Table 5), CM1-3 have a significantly reduced number of features than other SL classifiers published in the literature, testifying the lower complexity of our models. For instance, despite the reduced complexity, CM1 has an accuracy equal to about 92% of that of [30], but exploiting only one tenth of the features used in Ref. [30]. Other SL classifiers published in the literature achieved intermediate accuracy levels between those obtained in Refs. [5,6,30]. As an example, ensemble discriminant sparse projections (EDSP) was successfully applied to the SL classification of the excerpts extracted from GTZAN with an accuracy of 85% and a number of features of 768 [31]. An analogous number of features was more recently exploited in Ref. [32] via linear subspace dimensionality reduction (LSDR) techniques, in this case enhancing the accuracy to 91%. Finally, a topology preserving non-negative tensor factorization (TPNTF) method reported 94% accuracy by using 7680 audio features, the largest number of features with respect to all other SL classifiers.

Comparisons described in this section suggest that, while forced to work as single-genre classifiers, our models have slightly

worse performances compared to other SL approaches, even if the number of exploited features is significantly reduced. This is not surprising as our models are not conceived to work as high-accuracy SL classifiers. However, relatively high values of accuracy are achieved by CM1-3, testifying the capabilities of our models to meaningfully preserve the original SL information of GTZAN. In this framework, the aim of the newly developed models is not that of substituting other well-established SL approaches, but instead that of providing complementary information such as the ML classification and the fuzzy degree of belonging to music genres.

We must note at this point that a more substantial test should be performed against other ML models. To this end, Section 5.1 is concerned with a detailed comparison of CM1-3 to other ML classifiers previously published in the literature.

5.1. Detailed comparison with other multi-label classifiers

The comparison with other state-of-the-art ML approaches is a crucial point to test the capabilities of our models. To this end, we have quantitatively analyzed the performances of the popular ML classifier DS [13]. DS is a multi-genre MGR tool capable to quantify the belonging of music content to 10 music genres with a normalized coefficients vector. Output vector normalization represents one of the most important differences between DS and CM1-3. Our models, indeed, do not contain any a priory assumption on the amplitude of the output vector. This allows for a larger informative content to be returned. We can clarify this statement with the following example. Let us consider, as a limit condition, a possible music track with a full degree of belonging to simultaneously all the 10 genres considered in this analysis. The evaluation of this song by means of a non-normalized classification model like CM1-3 should result in a large output for each of the classifiers. On the contrary, a normalized MGR model would return a degree of belonging of 10% for all classifiers, carrying an information non-significant to even low-threshold applications and almost completely losing the original information. Another substantial difference between our models and DS is their optimized implementation. DS relies on Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) Networks, while our models are uniquely based on simple analytical formulas with reduced number of features. DS has an accuracy, evaluated with the prescriptions discussed above, of 74%, see Table 5. This value is similar to those obtained with our classification models. DS exploits all the 518 features discussed in Section 4.2. Also in this case, our models introduce a noticeable simplification in the number of input features, made possible by the native feature selection capabilities of BP. For instance, the features exploited by CM2, which has the same average accuracy of DS, are only about 11% than those used by DS.

A more detailed comparison of the capabilities of DS and CM1-3 can be performed by a case-by-case analysis of the content of GTZAN. For simplicity, we ordered all GTZAN audio files with respect to the Euclidean norm of the coefficients obtained with CM3, i.e. $\|\mathbf{c}^{(i)}\| = \sqrt{\sum_j c_j^{(i)}}$. More precisely, let us indicate with $S = \{I_1, \ldots, I_{1000}\}$ the vector of indexes of the GTZAN audio files ordered so that $\|\mathbf{c}^{(I_1)}\| \geq \|\mathbf{c}^{(I_2)}\| \geq \cdots \geq \|\mathbf{c}^{(I_{1000})}\|$ and with S_k the subset of the first k elements of S, i.e. $S_k = \{I_1, \ldots, I_k\}$. In this way, one can expect that each S_k subset contains a partition of the original GTZAN dataset with increasing consistency (i.e. better correlation between input features and underlying genre(s), according to CM3) for decreasing k-values. The corresponding accuracy was evaluated on several subsets S_k.

In Fig. 7 we reported *Acc* values for DS, CM1 and CM3 evaluated on different S_k subsets, with k ranging from 100 to 1000 (using a step of 50). As expected because of the adopted ordering, the trend of the accuracy for all considered models is monotonically descending for increasing k-values, indicating that the output of CM3 represents a good estimate for the quality of input features. Additionally, CM1 (open circles) has the highest accuracy when considering the subsets from S_{100} to S_{600} (i.e. the first 600 songs ordered with our method). In the same range, DS (reversed triangles) and CM3 (open crosses) have similar accuracy. Only for values of k greater than 750 DS performs

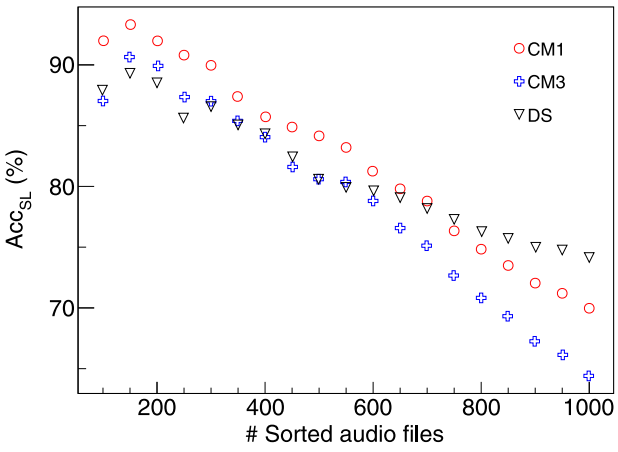

Fig. 7. Accuracy of CM1 (open circles), CM3 (open crosses) and DS (reversed triangles), evaluated with the metric described by Eq. (5) over different S_k subsets of the GTZAN dataset.

Table 6
S_{100} excerpts classified in a different manner by CM3 and DS.

Audio file	CM3			DS		
hiphop.00075	Pop (0.90)	HipHop (0.60)	Reggae (0.27)	HipHop (0.80)	Pop (0.08)	Reggae (0.04)
hiphop.00014	Metal (0.93)	HipHop (0.37)	Disco (0.25)	Reggae (0.79)	Rock (0.05)	Disco (0.05)
hiphop.00095	Metal (0.97)	HipHop (0.37)	Disco (0.30)	Metal (0.91)	HipHop (0.03)	Disco (0.02)
hiphop.00047	HipHop (1.0)	Pop (0.313)	Disco (0.21)	Pop (0.40)	HipHop (0.37)	Reggae (0.06)
hiphop.00084	HipHop (0.90)	Metal (0.56)	Rock (0.07)	Reggae (0.28)	HipHop (0.21)	Jazz (0.17)
country.00067	Country (1.0)	HipHop (0.18)	Rock (0.13)	Jazz (0.66)	Blues (0.19)	Country (0.07)
country.00064	Classical (0.86)	Country (0.63)	Rock (0.07)	Jazz (0.70)	Classical (0.12)	Blues (0.11)
country.00071	Country (1.0)	Classical (0.33)	Jazz (0.23)	Classical (0.55)	Country (0.28)	Jazz (0.07)
country.00014	Pop (0.96)	Disco (0.35)	Country (0.16)	Disco (0.34)	Rock (0.25)	Pop (0.17)
country.00069	Blues (0.91)	Country (0.44)	Classical (0.09)	Country (0.77)	Rock (0.07)	Blues (0.07)
blues.00031	Blues (1.0)	Classical (0.63)	Country (0.40)	Country (0.55)	Blues (0.29)	Rock (0.11)
pop.00013	Country (0.98)	Reggae (0.71)	Rock (0.10)	Pop (0.30)	HipHop (0.28)	Reggae (0.13)
jazz.00010	Country (1.0)	Jazz (0.51)	Rock (0.17)	Jazz (0.60)	Classical (0.22)	Country (0.09)
jazz.00006	Reggae (0.81)	Jazz (0.72)	Rock (0.10)	Jazz (0.80)	Blues (0.13)	Country (0.03)
classical.00054	Classical (0.92)	Reggae (0.51)	HipHop (0.33)	Country (0.45)	Classical (0.18)	reggae (0.12)
classical.00089	Classical (1.0)	Blues (0.27)	Metal (0.03)	HipHop (0.51)	Classical (0.16)	Metal (0.15)
classical.00079	Blues (1.0)	Classical (0.52)	Country (0.06)		Classical (0.99)	
reggae.00076	Pop (0.86)	Reggae (0.64)	Country (0.09)	Reggae (0.49)	HipHop (0.46)	Rock (0.02)
reggae.00086	HipHop (1.0)	Reggae (0.37)	Blues (0.32)	Disco (0.72)	Pop (0.15)	HipHop (0.04)
disco.00026	Pop (0.976)	Reggae (0.22)	HipHop (0.18)	Disco (0.77)	Pop (0.08)	HipHop (0.08)

$$
\begin{aligned}
blues: \quad & f_1 = -1.3486 \cdot erf\left(erf\left(\min\left(chstftS_4 + 0.013941 \cdot scontL_5 \cdot mfccA_{17} \cdot chstftS_4 + \right.\right.\right. \\
& + \left.\left.\left. 3.5853 \cdot sflatG_1 + e^{mfccA_{17}} + -1.4119 \cdot (-1.9848 \cdot chcensS_3 + chcqtA_{12}) + chstftL_6 \cdot mfccS_2 + \frac{sflatG_1}{chcensG_7} + mfccW_7, \Omega\left(scontL_5\right)\right)\right)\right) \\
classical: \quad & f_2 = 0.99971^{srollG_1 \cdot \sinh\left(mfccW_4 + \frac{mfccG_4}{scontL_4}\right)} \cdot srollG_1{}^{srollS_1} \\
country: \quad & f_3 = -0.2973 \cdot tonzS_3 \cdot (0.38316 + chstftA_6 + tonzA_2 \cdot mfccL_8 + tonzG_4 \cdot mfccM_{17} \cdot tonzL_4) \cdot scontA_1 \cdot tonzL_3 \\
disco: \quad & f_4 = \min\left(\frac{chcensL_2}{scentS_1 \cdot \max\left(mfccA_9, chcensA_2\right)}, 1.6088 \cdot scentS_1 \cdot \right. \\
& \cdot \left. 0.29757^{(0.96941 + |\max(mfccK_{11}, mfccA_1)| + -1.0684 \cdot (mfccW_7 + -1 \cdot mfccK_4))} \cdot \max\left(\sinh\left(spbwM_1\right) + -7.7223 \cdot scontW_2, chcensL_2\right)\right) \\
hiphop: \quad & f_5 = \left(2.2572 + spbwW_1\right) \cdot \min\left(\arctan\left(chcqtA_8\right), \Omega\left(\ln\left(chstftM_{11}\right)\right)\right)^{\frac{1.4129 \cdot \min(srollL_1, zcrM_1)}{zcrS_1}} \cdot \left(\tanh\left(chcqtA_{10}\right) + -0.50974 \cdot \max\left(chcensW_6, chcensW_5\right) + \tanh\left(scontW_4\right)\right) \\
jazz: \quad & f_6 = \Omega\left(0.16454 \cdot mfccS_{10} + scontK_4 + scontK_5 + \right. \\
& + \left. \Omega\left(-15.448 \cdot chstftW_2 \cdot \Omega\left(0.25913 \cdot mfccS_{10} + scontK_4 + \min\left(scontK_5, mfccW_{15}\right) + \max\left(mfccK_3, chstftA_{10}{}^{\sin(chstftW_5) \cdot chstftW_2 \cdot chstftW_{12}}\right)\right) \cdot chstftW_5\right)\right) \\
metal: \quad & f_7 = \max\left(7.5231 \cdot \min\left(mfccL_4, 0.12811 \cdot \sin\left(-0.68361 \cdot \left(chcensL_5{}^{chstftS_5} + scontA_5\right)\right)\right) + -7.4118 \cdot chcqtS_4 \cdot e^{mfccA_{17}}, \Omega\left(mfccL_{14} + mfccK_4\right)\right) \\
pop: \quad & f_8 = \tanh\left(0.00029 \cdot mfccG_{20} \cdot \max\left(0.022887, mfccA_9 + -1.0283 \cdot scontW_7 \cdot mfccA_{11}\right) \cdot spbwM_1{}^{spbwM_1} \cdot |scontW_7|\right) \\
reggae: \quad & f_9 = 0.81297 \cdot \Omega\left(srollL_1 + mfccK_1 + \max\left(\max\left(chstftK_{12}, chstftW_4\right) \cdot chstftW_{12}, chcqtK_7\right) + chcensA_7 + mfccW_{16}\right) \cdot ||\min\left(mfccW_{16}, chstftL_{10}\right)|^{\max(mfccW_{16}, chstftK_1) \cdot chstftW_3|} \\
rock: \quad & f_{10} = 0.60571^{\left(-2.045 + \max\left(1.3533^{(chstftK_{10} + tonzK_2 + \max(mfccS_{12} + -1 \cdot scentW_1 + chstftW_5 \cdot \sin(\sinh(scontW_7)) \cdot scentA_1, \tan(chcensK_{12}) + mfccA_{14}))}, \frac{mfccM_{20}}{chstftS_8}\right)\right)}
\end{aligned}
$$

Fig. A.8. Analytical formulas of CM1.

$$
\begin{aligned}
blues: \quad & f_1 = -0.039067 \cdot \max\left(chcensL_{12}, -0.34335 \cdot mfccW_{17}\right) \cdot mfccL_7 \cdot \left(chstftW_2 + \cos\left(srollG_1\right)\right) \\
classical: \quad & f_2 = \tanh\left(scontL_5 \cdot \Omega\left(\frac{mfccG_6}{1.2344^{scontL_5}}\right)\right)^{\frac{sflatG_1}{zcrM_1}} \\
country: \quad & f_3 = |-0.77777 + (-1.5717 + erf\left(mfccM_{20}\right) + mfccW_{20}) \cdot tonzS_4 \cdot mfccM_{17} + 1.5119 \cdot tonzS_3 \cdot scontW_1 \cdot mfccS_5| \cdot tonzS_3 \\
disco: \quad & f_4 = \sin\left(1.2497 \cdot \left(0.23953 + tonzW_2 + scontM_3{}^{mfccW_2} + \sin\left(-0.97047 \cdot scontM_3\right)\right) \cdot chcensL_2 \cdot \left(\cos\left(0.41994 + scentG_1\right) + chstftG_7 + -0.68185 \cdot tonzK_6\right)\right) \\
hiphop: \quad & f_5 = 2.8048 \cdot \max\left(\sin\left(\frac{35.821}{mfccS_5}\right), chcqtL_{10}\right) \cdot \sin\left(3.6393 \cdot chstftM_{12}\right) \cdot \left(scontW_4 + chcqtL_{10}\right) \cdot chstftM_{12} \\
jazz: \quad & f_6 = \sin\left(\max\left(\frac{-0.19811 \cdot (scontK_4 + scontK_5 + -0.067264 \cdot \min\left(-0.25723, mfccM_{17}\right)) \cdot chstftW_{12} \cdot \sin\left(0.99973 \cdot \cosh\left(chstftW_{10}\right) \cdot \sin\left(chcensS_8 + chstftW_{10} + mfccW_{18}\right) \cdot \sin\left(chstftW_{10}\right)\right)}{zcrG_1}, \right.\right. \\
& - \left.\left. 0.013281\right)\right) \\
metal: \quad & f_7 = \tanh\left(-6.9271 \cdot chstftL_{12} \cdot mfccM_{13} \cdot zcrL_1 \cdot \Omega\left(1.5684 + \cos\left(scontM_5\right) + mfccW_{10}\right) \cdot mfccA_4 \cdot scontW_5\right) \\
pop: \quad & f_8 = \min\left(chcqtG_{12}, 1.3302 \cdot scentS_1 \cdot sflatS_1 \cdot \max\left(0.20382, -1.0802 \cdot mfccM_{11} + mfccM_9\right) \cdot \left(spbwA_1{}^{scontW_7} + chstftK_5\right)\right) \\
reggae: \quad & f_9 = \min\left(0.003641, chcensL_{10}\right) \cdot \arccos\left(\sin\left(\tan\left(\arccos\left(\sin\left(mfccK_{17} + srollM_1\right)\right)\right)\right)\right) \cdot 118.25^{\Omega(\max(\max(mfccK_1, mfccK_{10} + mfccW_{17} + srollL_1), chcensL_{10}))} \\
rock: \quad & f_{10} = e^{\cos\left(mfccS_{12}\right)} \cdot e^{\cos\left(scontA_5\right)} \cdot 2.0993E{-}05^{\max(chcqtS_4, \max(mfccM_{20}, mfccA_{13}))}
\end{aligned}
$$

Fig. A.9. Analytical formulas of CM2.

$$
\begin{aligned}
blues: \quad & f_1 = \max\left(0.035614, -0.038024 \cdot mfccA_6 \cdot chstftW_5 \cdot mfccW_7\right) \\
classical: \quad & f_2 = erf\left(1E{-}06 \cdot e^{scontL_5}\right)^{srollS_1} \\
country: \quad & f_3 = \sin\left(\frac{scontL_1}{mfccG_{20}}\right) \cdot scontG_1 \cdot tonzS_3 \cdot tonzS_4 \\
disco: \quad & f_4 = \frac{\min\left(scentS_1, srollL_1\right) \cdot chcensL_2}{chcensA_3} \\
hiphop: \quad & f_5 = 0.003772 \cdot scontW_4 \cdot mfccS_5 \cdot chstftM_9 \cdot mfccG_{11} \\
jazz: \quad & f_6 = \max\left(-0.67926 \cdot (scontK_5 + scontK_4) + chcensM_9 + -0.026954 \cdot mfccS_{10} \cdot chstftM_{10} \cdot mfccS_9, -1.2E{-}05\right) \\
metal: \quad & f_7 = \Omega\left(scontS_5 + -0.020284 \cdot mfccA_4 + \cos\left(scontA_5\right)\right) \\
pop: \quad & f_8 = \tanh\left(1.8376 \cdot zcrW_1 \cdot \max\left(0.19917, 0.65209 + mfccA_{17} + mfccA_9 + -1 \cdot mfccA_{11}\right) \cdot sflatA_1\right) \\
reggae: \quad & f_9 = \min\left(\max\left(chcensL_{10}, srollL_1 \cdot mfccM_{15}\right), sflatG_1\right)^{\max(\max(mfccK_1, mfccW_{16} + srollL_1), sflatG_1)} \\
rock: \quad & f_{10} = 0.75409^{mfccS_{12}}
\end{aligned}
$$

Fig. A.10. Analytical formulas of CM3.

better than our models. This is a reasonable fact, as, due to their limited complexity, our models have the tendency to be more conservative for music content identified by features poorly correlated with the corresponding music genre(s) of belonging.

Finally, we carried out a detailed analysis of all songs in the subset S_{100}. Within this subset, the classifications given by CM3 and DS agree to each other and with the ground truth of GTZAN for 80 songs. In Table 6, we report the classification obtained with CM3 and DS for the remaining 20 songs, for which the genre resulting in the largest output coefficient is not the same for the two models. More in detail, Table 6 reports the most significant 3 coefficients obtained with CM3 and DS, sorted from the largest to the smallest from left to right. In such a way, the leftmost coefficient for each of the considered models corresponds to that of the dominant genre to be compared with the ground truth of GTZAN. By inspecting the output of DS, one can easily observe that there are 8 songs for which the dominant genre corresponds to that indicated by GTZAN. DS therefore correctly classifies $80 + 8 = 88$ songs of S_{100}. In the case of CM3, there are 7 songs whose classification, in the sense of SL, agrees with GTZAN. We can therefore state that the accuracy achieved by CM3 on S_{100} (87%) is not significantly different from the one of DS (88%). However, through a careful inspection of all audio files of S_{100}, we evidenced some interesting facts that we point out in the following.

The file hiphop.00014 results to be mislabeled. It is an excerpt of the song *Fight For Your Right* of the Beastie Boys group; the genre of this song, erroneously classified by GTZAN as Hip-hop, is mainly Hard Rock, even though a small Pop Rock component is also present, as reported also by Wikipedia [34]. It is worth observing that CM3 classifies this song mainly as Metal, which is a genre with a reasonably large degree of superposition with the Hard Rock genre and also the most similar genre among the 10 genres considered by the GTZAN dataset. Additionally, CM3 correctly suggests that the Hip-hop genre is present but it is not the dominant genre (the Hip-hop function returns a value of 0.37 vs 0.93 returned by the Metal function). On the contrary, if one looks at the output of DS for this song, reported in Table 6, a dominant Reggae component is returned (79%), in contrast with the commonly accepted classification for this song.

The file country.00014 is an excerpt of the song *Silver Threads and Golden Needless* singed by Nina Martinique. This song was originally recorded by The Springfields, a Folk/Pop group. Its classification in GTZAN is Country. CM3 mainly associates this song to Pop (0.96), while country is also present in minor part (0.16). According to DS, this song almost equally Disco (34%) and Rock (25%). Also in this case the output of CM3 is more reasonable than that of DS.

The song pop.00013 is an excerpt from *Every Time I Close My Eyes* singed by Babyface. This song is erroneously classified by GTZAN as Pop. Its dominant genre is R&B, as reported for example in Ref. [35]. In this case both DS and CM3 return as output more genres with similar amplitude. In particular, DS classifies this song as a combination of Pop (30%) and HipHop (28%), while CM3 classifies the same song as mainly Country (0.98) and Reggae (0.71). To our opinion it is not simple to state which combination better represents the R&B genre.

disco.00026 is an excerpt of the song *(Baby) Do the Salsa* by La Toya Jackson. This song is classified by Discogs with several genres [36], mainly Electronic and Pop among others. Moreover, the song exhibits a style quite similar to that of Madonna, an iconic singer of the pop genre. For these reasons, the genre Pop identified by CM3 could be considered a reasonable classification. DS associates this song predominantly to Disco (77%).

The number of songs that are correctly identified by CM3 further increases if we consider the possibility of CM3 to be used as multi-genre classifier. In fact, as it is evident in Table 6, for most songs the ground truth genre indicated by GTZAN is among the dominant genres suggested by CM3.

6. Conclusions

Music genre recognition tools for automatic classification of music content are highly demanded by modern music collection services such as Spotify and iTunes. In this framework, we have developed three multi-genre classification models capable to evaluate the degree of belonging of audio tracks to ten different music genres, accounting for the well-known existence of *fuzzy-like* boundaries between genres. The innovative aspects of our approach consist in the analytical implementation, the strongly reduced number of features and the absence of a priory assumptions on the output vector amplitude (i.e. degrees of belonging are fuzzy-like and it is not required for their sum to lead to 100%, with a consequent larger informative content carried by the output). Our models are derived by exploiting the unique capabilities of the Brain Project, a neuro-genetic distributed tool for the formal modeling of data, and the informative content contained in the GTZAN dataset, the most widely used collection for music genre recognition.

A careful comparison of the performances of our models against those of state-of-the-art tools previously published in the literature indicates that, even when forcing our models to work as single-label classifiers, their accuracy is only slightly lower than that of other single-label classifiers.

Despite the simple implementation and the limited number of features, the accuracy of our models turns out to be similar to that of other multi-label approaches based on Convolutional Neural Networks and Long Short-Term Memory Networks. In addition, our lower complexity model exploits only 38 of the 518 features typically used by multi-genre classification tools published in the literature.

Finally, the low computational complexity deriving from the analytical implementation and the reduced number of features makes our newly developed models particularly suitable to be integrated in low-cost embedded systems for real-time applications.

CRediT authorship contribution statement

G. Campobello: Formal analysis, Data curation, Writing-review and editing. **D. Dell'Aquila:** Formal analysis, Conceptualization, Writing - original draft, Writing-review and editing. **M. Russo:** Supervision, Conceptualization, Validation, Methodology, Software, Writing-review and editing. **A. Segreto:** Formal analysis, Data curation, Writing-review and editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Analytical formulations of our classification models

The reader can refer to this section for the full analytical representation of the classification models CM1-3 derived via BP. Related formulas can be easily implemented for comparisons with future approaches and/or for the integration of the classification model(s) described in this paper in applications aimed at multi-genre classification of music content via the audio features described in Section 4.2.

References

[1] Y. Panagakis, C. Kotropoulos, G. Arce, Music genre classification via sparse representations of auditory temporal modulations, in: Proc. the 17th European Signal Processing Conference, 2009, pp. 1–5.

[2] Y. Panagakis, C. Kotropoulos, Music genre classification via topology preserving non-negative tensor factorization and sparse representations, in: Proc. the 35th IEEE International Conference on Acoustics, Speech, and, Signal Processing, 2010, pp. 249–252.

[3] Y. Panagakis, C. Kotropoulos, G. Arce, Non-negative multilinear principal component analysis of auditory temporal modulations for music genre classification, IEEE Trans. Audio Speech Lang. Process. 18 (3) (2010) 576–588.

[4] C. Kotropoulos, G. Arce, Y. Panagakis, Ensemble discriminant sparse projections applied to music genre classification, in: Proc. the 20th International Conference on, Pattern Recognition, 2010, pp. 822–825.

[5] Y.-F. Huang, S.-M. Lin, H.-Y. Lu, Y.-S. Li, Music genre classification based on local feature selection using a self-adaptive harmony search algorithm, Data Knowl. Eng. 92 (2014) 60–76.

[6] R. Sarkar, S. Saha, Music genre classification using EMD and pitch based feature, in: Proceedings of Eighth International Conference on Advances in Pattern Recognition, ICAPR, IEEE, 2015, pp. 1–6.

[7] N. Scaringella, G. Zoia, D. Mlynek, Automatic genre classification of music content: a survey, IEEE Signal Process. Mag. 23 (2) (2006) 133–141, http://dx.doi.org/10.1109/MSP.2006.1598089.

[8] S. Lippens, J.P. Martens, T.D. Mulder, A comparison of human and automatic musical genre classification, in: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 4, 2004, pp. 233–236.

[9] K. Seyerlehner, G. Widmer, G. Peeters, Content-based music recommender systems: Beyond simple frame-level audio similarity dissertation zur erlangung des akademischen grades, 2011.

[10] H. Pálmason, B.þ. Jónsson, M. Schedl, P. Knees, Music genre classification revisited: An in-depth examination guided by music experts, in: Proceedings the 13th International Symposium on CMMR, Atosinhos, Portugal, 2017.

[11] GTZAN dataset, http://www.eecs.qmul.ac.uk/~sturm/research/GTZANtable2/index.html.

[12] B.L. Sturm, Classification accuracy is not enough, J. Intell. Inf. Syst. 41 (3) (2013) 371–406.

[13] P. Kozakowski, B. Michalak, Deep sound: Music genre recognition, 2016.

[14] M. Russo, A distributed neuro-genetic programming tool, Swarm Evol. Comput. 27 (2016) 145–155, http://dx.doi.org/10.1016/j.swevo.2015.10.009.

[15] M. Russo, A novel technique to self adapt parameters in parallel/distributed genetic programming, Soft Comput. (2020) http://dx.doi.org/10.1007/s00500-020-04982-w.

[16] R. Poli, W. Langdon, N. McPhee, A Filed Guide to Genetic Programming, Lulu Enterprises, UK Ltd., 2008.

[17] W. Stallings, Data and Computer Communications, Prentice Hall, Upper Saddle River, New Jersey, USA, 2006.

[18] Y. Chathuranga, K. Jayaratne, Automatic music genre classification of audio signals with machine learning approaches, GSTF J. Comput. (JoC) 3 (2) (2018).

[19] Y. Costa, L. Oliveira, C.N. Silla Jr., An evaluation of convolutional neural networks for music classification using spectrograms, Appl. Soft. Comput. 52 (2017) 28–38.

[20] L. Nanni, Y.C. MG, A. Lumini, M.Y. Kim, S.R. Baek, Combining visual and acoustic features for music genre classification, Expert Syst. Appl. 45 (2016) 108–117.

[21] M.L. Zhang, Z.H. Zhou, A review on multi-label learning algorithms, IEEE Trans. Knowl. Data Eng. 26 (8) (2014) 1819–1837.

[22] F. Mercury, M. Caballé, Barcelona, https://en.wikipedia.org/wiki/Barcelona_(Freddie_Mercury_and_Montserrat_Caball%C3%A9_song).

[23] C.M. Bishop, Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1996.

[24] J. Koza, Genetic Programming: On the Programming of Computers by Natural Selection, MIT Press, Cambridge, MA, USA, 1992.

[25] M. Russo, Distributed fuzzy learning using the MULTISOFT machine, IEEE Trans. Neural Netw. 12 (2001) 475.

[26] M. Russo, Genetic fuzzy learning, IEEE Trans. Evo. Comput. 4 (3) (2000) 259–273.

[27] B.L. Sturm, The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use, 2013, CoRR arXiv:abs/1306.1461.

[28] B. McFee, C. Raffel, D. Liang, D.P. Ellis, M. McVicar, E. Battenberg, O. Nieto, Librosa: Audio and music signal analysis in python, 2015.

[29] H. Bahuleyan, Music genre classification using machine learning techniques, 2018, arXiv preprint arXiv:abs/1804.01149.

[30] Y. Panagakis, C. Kotropoulos, G.R. Arce, Non-negative multilinear principal component analysis of auditory temporal modulations for music genre classification, IEEE Trans. Audio Speech Lang. Process. 18 (3) (2010) 576–588, http://dx.doi.org/10.1109/TASL.2009.2036813.

[31] C. Kotropoulos, G.R. Arce, Y. Panagakis, Ensemble discriminant sparse projections applied to music genre classification, in: 2010 20th International Conference on Pattern Recognition, 2010, pp. 822–825, http://dx.doi.org/10.1109/ICPR.2010.207.

[32] Y. Panagakis, C. Kotropoulos, G.R. Arce, Music genre classification via sparse representations of auditory temporal modulations, in: 2009 17th European Signal Processing Conference, 2009, pp. 1–5.

[33] Y. Panagakis, C. Kotropoulos, Music genre classification via topology preserving non-negative tensor factorization and sparse representations, in: 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, 2010, pp. 249–252, http://dx.doi.org/10.1109/ICASSP.2010.5495984.

[34] B. Boys, (You Gotta) Fight for Your Right (To Party!), https://en.wikipedia.org/wiki/(YouGotta)_Fight_for_Your_Right_(To_Party!).

[35] Babyface, Every Time I Close My Eyes, https://en.wikipedia.org/wiki/Every_Time_I_Close_My_Eyes.

[36] L.T. Jackson, Do The Salsa, https://www.discogs.com/composition/b1c59d7c-a310-44f8-bbc8-99e9536bd38d-do-the-salsa.