Article

A Two-Step Approach for Classifying Music Genre on the Strength of AHP Weighted Musical Features

Yu-Tso Chen ¹ , Chi-Hua Chen ^{2,*} , Szu Wu ³ and Chi-Chun Lo ³

¹ Department of Information Management, National United University, Miaoli 36003, Taiwan; yutso.chen@nuu.edu.tw

² College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China

³ Department of Information Management and Finance, National Chiao Tung University, Hsinchu 30010, Taiwan; szuwu@hotmail.com (S.W.); cclo@faculty.nctu.edu.tw (C.-C.L.)

* Correspondence: chihua0826@fzu.edu.cn; Tel.: +86-183-5918-3858

Received: 16 October 2018; Accepted: 21 December 2018; Published: 25 December 2018

Abstract: Music is a series of harmonious sounds well arranged by musical elements including rhythm, melody, and harmony (RMH). Since music digitalization has resulted in a wide variety of new musical applications used in daily life, the use of music genre classification (MGC), especially MGC automation, is increasingly playing a key role in the development of novel musical services. However, achieving satisfactory performance of MGC automation is a practical challenge. This paper proposes a two-step approach for music genre classification (called TSMGC) on the strength of analytic hierarchy process (AHP) weighted musical features. Compared with other MGC approaches, the TSMGC has three strong points for better performance: (1) various musical features extracted from the RMH and the calculated entropy are comprehensively considered, (2) the weight of features and their impact values determined by AHP are applied on the basis of the Exponential Distribution function, (3) music can be accurately categorized into a main-class and further sub-classes through a two-step classification process. According to the conducted experiment, the result exhibits an accuracy rate of 87%, which demonstrates the potential for the proposed TSMGC method to meet the emerging needs of MGC automation.

Keywords: music genre classification; feature extraction; symbolic music content; analytic hierarchy process; machine learning

1. Introduction

Music is an essential part of human civilization. Different types of music composed of various elements (e.g., rhythm, timbre, tempo.) are played on numerous and various occasions, depending on the situation. In recent years, the digitalization of music has dramatically extended the range of available music applications. According to [1], since users prefer to browse music by genres than other alternatives, the technique of differentiating the music types of genres (music genre classification, MGC) has become a popular tool of choosing appropriate music for an occasion or application. In particular, MGC automation attaches importance to the needs of dealing with a large amount of music within a specified time; for example, MGC is used for musical treatments [2].

Corrêa and Rodrigues [3] presented a review of the most important studies on MGC. In an effort to compare the pros and cons of the published research literature, they investigated the most common music features and classification techniques used. In the field of music information retrieval, MGC methods are generally categorized into those that analyze song lyrics or those that analyze musical content [4]. Although analyzing song lyrics is a straightforward way to identify the emotional expression of music [5–8], its performance is practically limited for songs with fewer lyrics. For this

reason, different emotional expressions by songs with the same lyric should be expressed and identified by varied rhythm, timbre, and tempo. In addition, this type of MGC cannot classify a song with nonsense lyrics or a scat style, or an instrumental piece. Besides that, audio data-based musical content, symbolic data-based musical content, community meta-data, or hybrid information are also considerable alternatives [9]. However in practice, most studies adopt music features based on music content to perform MGC, in particular for automatic MGC in a systematic manner [10,11]. Although both audio content-based and symbolic content-based representations have distinct advantages and drawbacks, the MGC approaches that rely on symbolic content-based features deliver better accuracy than those that rely on audio content-based features, because the symbolic data can offer higher-level music representation [3]. In addition to the issue of feature extraction, the design and choice of the classifier plays an essential role in the automatic classification of music [12,13]. MGC generally obtains musical features by analyzing the musical content with machine learning support [2,14–16]. In reality, these approaches require less computational time to evaluate a single feature, or to deal with musical compositions which have the significant characteristics of a specific music class (genre). According to [11], the common classifier for symbolic data-based MGC include k-nearest neighbor (kNN), support vector machines (SVM), Bayesian classifier, artificial neural networks (ANN), and self-organizing maps (SOM). The review article [3] summarized the most important investigations on symbolic data-based MGC and made a comparison on their classification approaches and the adopted features. For more details on this topic, the reader is invited to read this review article.

In theory, it might be difficult to make symbolic data-based MGC for classifying musical compositions with few differences among the evaluated features, or with too many genres to be considered simultaneously in a MGC operation. That is, the enhancement of feature extraction from symbolic data and the use of genre hierarchy are two considerable issues. In general, a music composition is composed of rhythm, melody, and harmony (RMH), and thus a reasonable focus for improving MGC would be feature factors associated with RMH [17,18].

Feature extraction is a critical step for MGC operation, it selects the music features that can reflect the significant and discriminative characteristics of different types of genres. As was addressed in [19], the choice of suitable features influences the success of MGC tasks, whereas the performance of the classifier usually relies on the selection of features. As feature extraction is required for the analysis of, and to elicit the RMH-related features, two critical concerns should be taken care. First, different RMH features represent or correspond to different effects of music playing; second, the influence of the weighting of the features might affect the MGC results. Accordingly, methodologies for factor analysis like the analytic hierarchy process (AHP) [20] would be useful for working with feature extraction methods, and to determine the weight of each feature in order to filter out improper features.

Furthermore, when music innovation and reformulation will result in more presentation styles of musical composition, it will be insufficient to classify music just according to a single level of music genres if the MGC accuracy is guarantee-required. In order to solve this problem, the design of a MGC approach should consider to use more genre levels; e.g., two levels, one for the main-classes and the other for the sub-classes.

The remainder of the paper is organized as follows. Section 2 presents the design concepts which include the entropy analysis method, the exponential distribution, AHP, and machine learning. Section 3 describes the proposed two-step MGC approach based on AHP weighted musical features (TSMGC). The practical experiment results and analyses are given in Section 4. Finally, the paper is concluded and remarks for future work are given in Section 5.

2. The Design Concepts

Based on the above introduction, this paper presents a novel approach capable of improving the MGC performance by combining the following design concepts.

2.1. Entropy Analysis Method

The entropy analysis method is applied to calculate the entropy of the analyzed data, and is used to generate an indicator of complexity as a feature factor for classification. From a realistic point of view, many effective solutions to real-world problems rely on non-linear approaches. Accordingly, non-linear entropy analysis methods, including approximate entropy (ApEn) and sample entropy (SampEn), are frequently applied as information gaining methods to obtain vital data features.

The ApEn method, a non-linear approach used to measure the complexity of data series, was proposed in 1991. ApEn is often adopted to present the complexity of time series data as a non-negative number which indicates the probability of a new message being created from the time series data. When the complexity of the time series is high, the ApEn value is large [21]. Suppose that the dataset of a time series consisting of N values is respectively recorded as $u(1), u(2), \dots, u(N)$; an m -dimensional matrix X , where m is the number of requested data, is defined. Therefore, $X(i)$ is denoted as $[u(1), u(2), \dots, u(i+m-1)]$, where $1 \leq i \leq N - m + 1$. the ApEn value, $\text{ApEn}(m, r, N)$, can be obtained by Equations (1) to (4). First, Equation (1) is used to calculate the distance between $X(i)$ and $X(j)$, marked as $d[X(i), X(j)]$. Next, let r be a threshold, Equation (2) can compute the value of $C_i^m(r)$ with the condition of $d[X(i), X(j)]$ less than or equal to r . Then, Equation (3) determines the value of the defined $\Phi^m(r)$. Finally, the ApEn value, $\text{ApEn}(m, r, N)$, can be obtained by Equation (4).

$$d[X(i), X(j)] = \max[|u(i+k) - u(j+k)|], \text{ where } 0 \leq k \leq m-1 \quad (1)$$

$$C_i^m(r) = \frac{\text{number of } X(j), \text{ where } d[X(i), X(j)] \leq r}{N - m + 1}, \text{ where } 1 \leq i, j \leq N - m + 1 \quad (2)$$

$$\Phi^m(r) = \frac{1}{N - m + 1} \left(\sum_{i=1}^{N-m+1} \log(C_i^m(r)) \right) \quad (3)$$

$$\text{ApEn}(m, r, N) = \Phi^m(r) - \Phi^{m+1}(r) \quad (4)$$

SampEn, proposed in 2000, is considered to be an enhanced ApEn. Compared with ApEn, SampEn has better classification performance, especially for accuracy and consistency [22]. SampEn also adopts Equations (1) and (2) to calculate the distance between $X(i)$ and $X(j)$, and the value of $C_i^m(r)$. However, the functions for computing $\Phi^m(r)$ and $\text{SampEn}(m, r, N)$ are given as Equations (5) and (6).

$$\Phi^m(r) = \frac{1}{N - m + 1} \left(\sum_{i=1}^{N-m+1} C_i^m(r) \right) \quad (5)$$

$$\text{SampEn}(m, r, N) = -\log \left(\frac{\Phi^{m+1}(r)}{\Phi^m(r)} \right) \quad (6)$$

2.2. Exponential Distribution

In order to estimate the impact value of the selected musical features in the later feature extraction operation, the use of the Exponential Distribution is made. The Exponential Distribution is a probability distribution that describes the time (or distance) between events that occur continuously and independently at a constant average rate. In probability theory, a probability density function (PDF) is a function whose value at any given sample in the specific sample space can be interpreted as providing a relative likelihood that the value of the random variable would equal that sample. The PDF value at two different samples can be used to infer how much more likely it is that the random

variable will equal to one sample compared with the other. The introduction of PDF can be found from web, like Wikipedia; the formulation of the PDF is described as Equation (7).

$$f(x; \lambda) = \begin{cases} \lambda e^{-\lambda x} & , x \geq 0 \\ 0 & , x < 0 \end{cases} \quad (7)$$

2.3. Analytic Hierarchy Process

According to Bishop [23], a feature is an individual measurable property or characteristic of a phenomenon being observed. Choosing informative, discriminating, and independent features is a crucial step for effective algorithms in pattern recognition and classification, but not all features contribute to the results. The analytic hierarchy process (AHP) proposed by Saaty [24] is a quantitative method capable of evaluating and analyzing multiple factors for decision-making applications through a meaningful and repeatable process. The use of AHP can help in selecting feature vectors that are helpful for the classification result, improving the classification accuracy and the operational efficiency of feature oriented classification applications.

2.4. Machine Learning

Machine learning is a field of computer science that uses statistical techniques to give computer systems the ability to progressively improve performance on a specific task with data, without being explicitly programmed. There are many machine learning techniques [25] applied to MGC, such as artificial neural network (ANN), k-nearest neighbors (kNN), support vector machine (SVM) [26] and deep learning [27]. Of these approaches, ANN and kNN are the most used in the MGC field.

ANN systems [25] are computing systems roughly based on the biological neural networks that constitute animal brains. An ANN is based on a collection of connected units or nodes called artificial neurons. Each connection between artificial neurons can transmit a signal from one to another. The artificial neuron that receives the signal can process it and then signal artificial neurons connected to it. In common ANN implementations the signal at a connection between artificial neurons is a real number, and the output of each artificial neuron is calculated by a non-linear function of the sum of its inputs. Artificial neurons and connections typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Artificial neurons may have a threshold such that only if the aggregate signal crosses that threshold is the signal sent. Typically, artificial neurons are organized in layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first (input), to the last (output) layer, possibly after traversing the layers multiple times.

kNN [25] is a simpler supervised learning algorithm in machine learning technologies. It is a non-parametric method frequently used for classification and regression. In kNN classification, the input consists of the k closest training examples in the feature space; the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If $k = 1$, the object is simply assigned to the class of that single nearest neighbor.

2.5. Design Highlights

When a machine learning based MGC method, like [2,14–16], selects one single factor to perform a classification operation, it will struggle to classify music into proper classes if the musical feature of the evaluated music compositions doesn't have significant differences. In order to effectively and efficiently perform MGC, the following design concepts should be considered. First, in input analysis, multiple features with reasonable weight-settings are necessary to differentiate the music. Second, a two-step process applying the hierarchical classes; i.e., more genre levels, will contribute to precisely categorize the examined musical composition into a 'correct' class.

3. The Proposed TSMGC Approach

This section introduces the proposed two-step MGC approach based on AHP weighted musical features (TSMGC). The core of the TSMGC approach is an AHP-supported classification model based on accessing several musical features of the music content. Before introducing the design details of the proposed TSMGC, the notations used for presenting the related operations, including musical content retrieval (MCR), musical content analysis (MCA), feature extraction (FE), and two-step classification (TSC) are defined in Table 1.

Table 1. The definition of parameters.

Parameter	Definition
N	The number of musical composition
n	The total number of classes
nm	The number of main classes
ns_i	The number of sub-classes belonging to the i th main class, where $1 < i < nm$
m	The number of features
nn^s	The number of musical notes in the s th musical composition
p_i^s	The pitch value of the i th musical note of the s th musical composition, where $1 < i < nn^s$
$i_{i,i+1}^s$	The musical interval between the i th musical note and the $(i+1)$ th musical note of the s th musical composition, where $1 < i < nn^s - 1$
p^s	The set of pitch of the s th musical composition
I^s	The set of musical intervals of the s th musical composition
C^s	The set of chords of the s th musical composition
R^s	The set of rhythms of the s th musical composition
E^s	The set of pitch entropy in the s th musical composition
f_i^s	The value of the i th feature of the s th musical composition, where $1 < i < m$
w_i^s	The weight of the i th feature of the s th musical composition, where $1 < i < m$

3.1. The Operation of the TSMGC

TSMGC includes three operating phases: model training, model testing, and music classification, as shown in Figure 1. In the model training stage, musical contents are firstly retrieved by MCR from respective training data. An MCA process then generates the features, including pitch, musical interval, chord, rhythm, and pitch entropy. Next, the weight of all the features is measured and determined by the FE method, which helps to build a machine learning-based MGC model. The purpose of model testing is to evaluate the performance of the constructed MGC model; the testing data (music) excluded from the training dataset are used for the MCR, MCA, FE, and TSC processes. If the testing result is satisfactorily accepted, the MGC model is confirmed; otherwise, the approach returns to the model training phase to build a new MGC model. The music classification phase executes MCR, MCA, FE and TSC based on the confirmed MGC model to categorize the music to be classified into a proper main class (main genre) and corresponding sub-classes (sub-genres).

Figure 1. The procedure of the two-step approach for music genre classification (called TSMGC) operation. MCR: musical content retrieval; MGC: music genre classification.

3.2. Musical Content Retrieval

The purpose of MCR is to retrieve the musical content (e.g., the set of musical notes) from the target data (music). In this study, jMusic [28] is used to analyze and retrieve the musical notes in MIDI (musical instrument digital interface) format. Each musical note consists of two musical elements, pitch and rhythm. The output of MCR is a set of numerical values transformed from the musical content of the target data in accordance with the musical elements; this is used in the MCA operation.

3.3. Musical Content Analysis

The MCA process measures the numerical values of the features including pitch, musical interval, chord, rhythm, and the entropy of the pitch. The process of determining the respective features for analysis is as follows.

- (1) **Pitch.** The pitch of each musical note can be generated as a numerical value according to the defined numerical value of a musical alphabet (as shown in Table 2). Any two notes that have the same pitch but different musical alphabets are seen as enharmonic equivalents. For instance, the notes as alphabet values C and B $\sharp$ are enharmonic notes, thus the transformed numerical values of these two notes are both equal to 1. Based on this rule, a total of 78 pitches can be referred in a music composition. The value of each feature depends on its frequency. Furthermore, since the highest and lowest pitch values are also considered, 80 values in terms of pitch-oriented features can be collected.
- (2) **Musical Interval.** An n-gram segmentation approach is applied to transform a note into a gram. Assuming the number of musical composition (N) equals 2, Equation (8) is used to measure the value of the musical interval between two successive notes. According to Wikipedia, the traditional musical theory has defined basic musical intervals, but the influence by semitone and

enharmonic notes should also be considered. The use of a semitone may form an additional musical interval [29]; on the other hand, enharmonic notes may conduct an equivalent interval. For instance, the interval from C to D $\sharp$ is an augmented second (A2), while that from C to E $\flat$ is a minor third (m3). Since D $\sharp$ and E $\flat$ are enharmonic notes, the musical intervals of A2 and m3 are equivalent. Furthermore, the transformation from a compound interval into a simple interval can significantly simplify the identification of musical interval. Based on the mentioned rules, the numerical value of every musical interval can thus be calculated. For example, on a semitone basis, the interval from C (i.e., the pitch value is 1) to F $\sharp$ (i.e., the pitch value is 7) is an augmented fourth (A4), the numerical value of this interval is set to 6 (i.e., 7 minus 1 leaves 6). Table 2 shows the numerical value of the defined musical intervals. The value of each musical interval feature is the number of times it appears in the musical composition. As a result, 12 musical interval-oriented features are determined.

$$p_{i,i+1}^s = |p_i^s - p_{i+1}^s| \quad (8)$$

- (3) **Chord.** The set of pitches from a musical composition can be used to determine its tonality type by comparison with tonality characteristics. Each music has a different set of chords according to its tonality; there are seven basic chords in a tonality. Take the 12 Variations on “Ah, vous dirai-je, Maman” by Mozart as an example; since it contains no rising-falling tone (as the symbols shown in Figure 2), the tonality of this musical composition is perceived as a major scale based on C. In the first measure, only chord I is included. The second measure contains chords IV and I, the third measure contains chords vii and I, and the fourth measure contains chords V and I. The number of times each chord appears in a musical composition is recorded as the value of the respective chord-oriented features. In this part, 7 chord-oriented features can be generated.
- (4) **Rhythm.** According to the Oxford English Dictionary II, a rhythm generally means a “movement marked by the regulated succession of strong and weak elements, or of opposite or different conditions.” In this study, 3 rhythm-related features are defined to record the lowest note value (i.e., the shortest duration), the highest note value (i.e., the longest duration), and the average note value of the musical composition.
- (5) **Entropy of the pitch.** The ApEn and SampEn methods are adopted in this analysis. First, the pitches of a musical composition are transformed into a time series. The data set of this time series is the core material for entropy computation. If 3 is chosen as the threshold and 2 as the number of dimensions, then ApEn(2, 3, nn^s) and SampEn(2, 3, nn^s) can be calculated as the pitch entropies.

Figure 2. The chord mapping in the first four-measures of 12 Variations on “Ah, vous dirai-je, Maman” by Mozart.

In summary, a dataset $F^s = \{P^s, I^s, C^s, R^s, E^s\}$ composed of 104 values by 80 pitch related features (P^s), 12 musical-interval related features (I^s), 7 chord related features (C^s), 3 rhythm related features (R^s), and 2 pitch-entropy-based features (E^s) is now ready for the feature extraction process.

Table 2. The sample of numerical value for musical alphabet.

Musical Alphabet	Enharmonic Note	Numerical Value
C	B $\sharp$	1
C $\sharp$	D $\flat$	2
D	-	3
D $\sharp$	E $\flat$	4
E	F $\flat$	5
F	E $\sharp$	6
F $\sharp$	G $\flat$	7
G	-	8
G $\sharp$	A $\flat$	9
A	-	10
A $\sharp$	B $\flat$	11
B	C $\flat$	12

3.4. Feature Extraction

The main purpose of feature extraction is to determine the weight (impact value) of each of the 104 features. Significant features can be determined according to their feature weights. In this process, the exponential distribution is assumed to describe the distribution of the feature values; the impact value of the features can be estimated by calculating the intersection area under two curves, which represents the value of the probability density functions (PDFs) for a feature in a selected class. Figure 3 shows an intersection area under two PDF curves for a feature in Classes 1 and 2, respectively. The solid line in Figure 3 represents the PDFs of Class 1, and the dashed line represents the PDFs of Class 2. The intersection point of these two PDF curves is X, and the shaded area denotes the intersection area under these two PDF curves. The smaller the intersection area, the more significant the feature. Conversely, a larger intersection area implies that the feature has less impact.

Figure 3. An intersection area under two probability density function (PDF) curves.

In order to estimate the intersection area, the PDFs of Classes 1 and 2 are defined as Equations (9) and (10), respectively. The mean of the feature values in Class 1 is $\frac{1}{\lambda_1}$, and that in Class 2 is $\frac{1}{\lambda_2}$. The intersection point x can be measured by Equation (11). Next, Equation (12) can be used to estimate the intersection area under these two PDF curves.

$$f(x; \lambda_1) = \lambda_1 e^{-\lambda_1 x} \quad (9)$$

$$f(x; \lambda_2) = \lambda_2 e^{-\lambda_2 x} \quad (10)$$

$$\begin{aligned}
f(x; \lambda_1) &= f(x; \lambda_2) \\
\Rightarrow \lambda_1 e^{-\lambda_1 x} &= \lambda_2 e^{-\lambda_2 x} \\
\Rightarrow \frac{\lambda_1}{\lambda_2} &= e^{(-\lambda_2 x) - (-\lambda_1 x)} = e^{x(\lambda_1 - \lambda_2)} \\
\Rightarrow \ln \frac{\lambda_1}{\lambda_2} &= x(\lambda_1 - \lambda_2) \\
\Rightarrow x &= \frac{\ln \frac{\lambda_1}{\lambda_2}}{\lambda_1 - \lambda_2}
\end{aligned} \tag{11}$$

$$\begin{aligned}
A &= \int_{x=0}^{\frac{\ln \frac{\lambda_1}{\lambda_2}}{\lambda_1 - \lambda_2}} f(x; \lambda_1) dx + \int_{x=\frac{\ln \frac{\lambda_1}{\lambda_2}}{\lambda_1 - \lambda_2}}^{\infty} f(x; \lambda_2) dx \\
&= \int_{x=0}^{\frac{\ln \frac{\lambda_1}{\lambda_2}}{\lambda_1 - \lambda_2}} (\lambda_1 e^{-\lambda_1 x}) dx + \int_{x=\frac{\ln \frac{\lambda_1}{\lambda_2}}{\lambda_1 - \lambda_2}}^{\infty} (\lambda_2 e^{-\lambda_2 x}) dx \\
&= \left(-e^{-\lambda_1 x} \right) \Big|_{x=0}^{\frac{\ln \frac{\lambda_1}{\lambda_2}}{\lambda_1 - \lambda_2}} + \left(-e^{-\lambda_2 x} \right) \Big|_{x=\frac{\ln \frac{\lambda_1}{\lambda_2}}{\lambda_1 - \lambda_2}}^{\infty} \\
&= 1 - e^{-\lambda_1 \frac{\ln \frac{\lambda_1}{\lambda_2}}{\lambda_1 - \lambda_2}} + e^{-\lambda_2 \frac{\ln \frac{\lambda_1}{\lambda_2}}{\lambda_1 - \lambda_2}}
\end{aligned} \tag{12}$$

Next, AHP is applied to adjust the feature weights as well as to rank the features by analyzing the number of records in each class and the weight of each feature. Given a set of n compared attributes denoted as $\{a_1, a_2, \dots, a_n\}$, and a set of weights denoted as $\{w_1, w_2, \dots, w_n\}$, the matrix M and W for the AHP computation can be built using Equation (13) [21].

$$\begin{aligned}
M &= \begin{bmatrix} a_{1,1} & \cdots & a_{1,j} & \cdots & a_{1,n} \\ \vdots & & \vdots & & \vdots \\ a_{i,1} & \cdots & a_{i,j} & \cdots & a_{i,n} \\ \vdots & & \vdots & & \vdots \\ a_{n,1} & \cdots & a_{n,j} & \cdots & a_{n,n} \end{bmatrix} \\
&\cong \begin{bmatrix} \frac{w_1}{w_1} & \cdots & \frac{w_1}{w_j} & \cdots & \frac{w_1}{w_n} \\ \vdots & & \vdots & & \vdots \\ \frac{w_i}{w_1} & \cdots & \frac{w_i}{w_j} & \cdots & \frac{w_i}{w_n} \\ \vdots & & \vdots & & \vdots \\ \frac{w_n}{w_1} & \cdots & \frac{w_n}{w_j} & \cdots & \frac{w_n}{w_n} \end{bmatrix} = W
\end{aligned} \tag{13}$$

where $a_{i,j} > 0, a_{i,j} = \frac{1}{a_{j,i}}$, and

$$w_i = \sum_{v=1}^n a_{i,v} \times \frac{1}{\sum_{u=1}^n a_{u,v}}$$

In order to calculate the weight of each class (i.e., matrix M_G), Equation (14) which is derived from Equation (13) was used. Assume that a 2-combination of n classes forms q groups (i.e., $q = C_2^n = \frac{n(n-1)}{2}$), the matrix M_G can be measured by Equation (14) with the given g_i as the size of the i th class.

$$\begin{aligned}
 M_G &= \begin{bmatrix} \frac{(g_1+g_2)}{(g_1+g_2)} & \dots & \frac{(g_1+g_2)}{(g_i+g_j)} & \dots & \frac{(g_1+g_2)}{(g_{n-1}+g_n)} \\ \vdots & & \vdots & & \vdots \\ \frac{(g_i+g_j)}{(g_1+g_2)} & \dots & \frac{(g_i+g_j)}{(g_i+g_j)} & \dots & \frac{(g_i+g_j)}{(g_{n-1}+g_n)} \\ \vdots & & \vdots & & \vdots \\ \frac{(g_{n-1}+g_n)}{(g_1+g_2)} & \dots & \frac{(g_{n-1}+g_n)}{(g_i+g_j)} & \dots & \frac{(g_{n-1}+g_n)}{(g_{n-1}+g_n)} \end{bmatrix} \\
 &\cong \begin{bmatrix} \frac{w_{1,G}}{w_{1,G}} & \dots & \frac{w_{1,G}}{w_{j,G}} & \dots & \frac{w_{1,G}}{w_{q,G}} \\ \vdots & & \vdots & & \vdots \\ \frac{w_{i,G}}{w_{1,G}} & \dots & \frac{w_{i,G}}{w_{j,G}} & \dots & \frac{w_{i,G}}{w_{q,G}} \\ \vdots & & \vdots & & \vdots \\ \frac{w_{q,G}}{w_{1,G}} & \dots & \frac{w_{q,G}}{w_{j,G}} & \dots & \frac{w_{q,G}}{w_{q,G}} \end{bmatrix} = W_g
 \end{aligned} \tag{14}$$

where $r_{1,G} = (g_1 + g_2), r_{2,G} = (g_1 + g_3), \dots, r_{q,G} = (g_{n-1} + g_n),$

$$a_{i,j,G} = \frac{r_{i,G}}{r_{j,G}} > 0, a_{i,j,G} = \frac{1}{a_{j,i,G}}, \text{ and } w_{i,G} = \sum_{v=1}^q a_{i,v,G} \times \frac{1}{\sum_{u=1}^q a_{u,v,G}}$$

Similarly, Equation (15) which is also derived from Equation (13) can be used to calculate the weight of each feature (i.e., matrix M_K) in the k th group, where the k th group is defined as $A_{k,j}$ for the j th feature.

$$\begin{aligned}
 M_k &= \begin{bmatrix} \frac{(1-A_{k,1})}{(1-A_{k,1})} & \dots & \frac{(1-A_{k,1})}{(1-A_{k,i})} & \dots & \frac{(1-A_{k,1})}{(1-A_{k,m})} \\ \vdots & & \vdots & & \vdots \\ \frac{(1-A_{k,i})}{(1-A_{k,1})} & \dots & \frac{(1-A_{k,i})}{(1-A_{k,i})} & \dots & \frac{(1-A_{k,i})}{(1-A_{k,m})} \\ \vdots & & \vdots & & \vdots \\ \frac{(1-A_{k,m})}{(1-A_{k,1})} & \dots & \frac{(1-A_{k,m})}{(1-A_{k,i})} & \dots & \frac{(1-A_{k,m})}{(1-A_{k,m})} \end{bmatrix} \\
 &= \begin{bmatrix} \frac{r_{1,k}}{r_{1,k}} & \dots & \frac{r_{1,k}}{r_{i,k}} & \dots & \frac{r_{1,k}}{r_{m,k}} \\ \vdots & & \vdots & & \vdots \\ \frac{r_{i,k}}{r_{1,k}} & \dots & \frac{r_{i,k}}{r_{i,k}} & \dots & \frac{r_{i,k}}{r_{m,k}} \\ \vdots & & \vdots & & \vdots \\ \frac{r_{m,k}}{r_{1,k}} & \dots & \frac{r_{m,k}}{r_{i,k}} & \dots & \frac{r_{m,k}}{r_{m,k}} \end{bmatrix} \\
 &\cong \begin{bmatrix} \frac{w_{1,k}}{w_{1,k}} & \dots & \frac{w_{1,k}}{w_{i,k}} & \dots & \frac{w_{1,k}}{w_{m,k}} \\ \vdots & & \vdots & & \vdots \\ \frac{w_{i,k}}{w_{1,k}} & \dots & \frac{w_{i,k}}{w_{i,k}} & \dots & \frac{w_{i,k}}{w_{m,k}} \\ \vdots & & \vdots & & \vdots \\ \frac{w_{m,k}}{w_{1,k}} & \dots & \frac{w_{m,k}}{w_{i,k}} & \dots & \frac{w_{m,k}}{w_{m,k}} \end{bmatrix} = W_k
 \end{aligned} \tag{15}$$

where $r_{1,k} = (1 - A_{k,1}), r_{2,k} = (1 - A_{k,2}), \dots, r_{m,k} = (1 - A_{k,m}),$

$$a_{i,j,k} = \frac{r_{i,k}}{r_{j,k}} > 0, a_{i,j,k} = \frac{1}{a_{j,i,k}}, \text{ and } w_{i,k} = \sum_{v=1}^m a_{i,v,k} \times \frac{1}{\sum_{u=1}^m a_{u,v,k}}$$

Based on the above, the final weight matrix W_F can be calculated by using Equation (16). Thus, the impact value of every features (i.e., ω_k for the k th feature) can be determined. The results calculated by the above AHP related operations give a specific weighting value for each feature, and the importance of the classification results can be screened out to improve the MGC accuracy and reduce the time required for classification.

$$W_F = \begin{bmatrix} \sum_{v=1}^q \frac{w_{1,G}}{w_{v,G}} & \dots & \sum_{v=1}^q \frac{w_{i,G}}{w_{v,G}} & \dots & \sum_{v=1}^q \frac{w_{q,G}}{w_{v,G}} \end{bmatrix} \begin{bmatrix} \sum_{v=1}^m \frac{w_{1,1}}{w_{v,1}} & \dots & \sum_{v=1}^m \frac{w_{i,1}}{w_{v,1}} & \dots & \sum_{v=1}^m \frac{w_{m,1}}{w_{v,1}} \\ \vdots & & \vdots & & \vdots \\ \sum_{v=1}^m \frac{w_{1,i}}{w_{v,i}} & \dots & \sum_{v=1}^m \frac{w_{i,i}}{w_{v,i}} & \dots & \sum_{v=1}^m \frac{w_{m,i}}{w_{v,i}} \\ \vdots & & \vdots & & \vdots \\ \sum_{v=1}^m \frac{w_{1,q}}{w_{v,q}} & \dots & \sum_{v=1}^m \frac{w_{i,q}}{w_{v,q}} & \dots & \sum_{v=1}^m \frac{w_{m,q}}{w_{v,q}} \end{bmatrix} \quad (16)$$

$$= \begin{bmatrix} \omega_1 & \dots & \omega_k & \dots & \omega_m \end{bmatrix} \text{ where } \omega_k = \sum_{u=1}^q \left(\sum_{v=1}^q \frac{w_{u,G}}{w_{v,G}} \times \sum_{v=1}^m \frac{w_{k,u}}{w_{v,u}} \right)$$

3.5. Two-Step Classification

The core concept of this two-step classification (TSC) is that music can be grouped into not just a couple of main classes (e.g., classical music, popular music, and rock music) but also into expanded sub-classes (e.g., medieval music, baroque music, classical era music, romantic music, and modern music for classical music). For this purpose, a TSC collaborated MGC model is required. For training and testing such a MGC model, this study adopts both kNN and ANN [30]. In theory, there will be n_p models to be trained for classifying the main classes, and n_i models for classifying the sub-classes of the i th main class. Based on the confirmed MGC model, the TSC realizes the proper main class and further determines the corresponding reasonable sub-classes.

4. Experiments and Analyses

This section presents a TSMGC experiment with the details of experimental settings including tools, samples, evaluation factor, and the execution. Based on the experiment results, the performance of TSMGC is analyzed and compared with other MGC methods.

4.1. Tools for Experiment Implementation

The Java and Matlab languages were adopted to develop and implement this experiment. The Java-based jMusic [28] program (version 1.6.4) was applied to perform the MCR processes. Eclipse (version 4.4.2), an integrated development environment, was used to implement MCA functions capable of generating and accessing the values of the features associated to pitch, musical interval, chord, rhythm, and pitch entropy. Matlab (version R2014a) was adopted to implement the proposed FE and TSC operations.

4.2. Samples and the Classes

The main classes for music defined in this experiment include classical music, popular music and rock music. The classical music main class contains five sub-classes: medieval music, baroque music, classical era music, romantic music and modern music. The popular music main class consists of five sub-classes, pop 1960s music, pop 1970s music, pop 1980s music, pop 1990s music, and pop 2000s music. The rock music main class consists of the 1960s music, classic rock music, hard rock music, psychedelic rock music and 2000s music sub-classes. Table 3 describes the selected 141 samples with their class settings.

Table 3. The samples.

Main-Class	Sub-Class	Sample Size of Sub-Class	Sample Size of Main-Class	Sample Size in Total
Classical	Medieval music	11	51	141
	Baroque music	10		
	Classical era music	10		
	Romantic music	11		
	Modern music	9		
Popular	Pop 1960s music	12	48	
	Pop 1970s music	10		
	Pop 1980s music	7		
	Pop 1990s music	8		
	Pop 2000s music	11		
Rock	1960s music	6	42	
	Classic rock music	10		
	Hard rock music	8		
	Psychedelic rock music	10		
	2000s music	8		

4.3. The Evaluation Factor

The evaluation factor for this experiment is the accuracy rate computed by Equation (17). N is the total number of sample cases; MGC_c is the correct number of classifications. A higher result accuracy rate indicates better performance of the adopted MGC approach.

$$Accuracy(\%) = \frac{MGC_c}{N} \times 100 \quad (17)$$

4.4. The Experiment Execution

In this study, the data distribution of the extracted features is theoretically assumed to be an exponential distribution. In order to verify whether the data distribution of the extracted features is exponentially distributed, the chi-squared test was used to verify the goodness-of-fit between the real distribution of the sample data and the expected one. For example, Figure 4 shows the feature data distribution of Chord V based on 48 pieces of popular music. The circle points denote the PDF of practical data distribution (Real), and the rectangle points denote the PDF of exponential distribution (Exp). Because the test results are $\chi^2 = 3.24 < \chi^2_{n-1=19} = 30.149$ when $\alpha = 0.05$ [31], no significant difference was observed; this implies that it follows an exponential distribution.

Figure 4. The PDF example of feature data distribution, Chord V.

In terms of arranging the training data and test data, a k -fold cross-validation method was used [26]. When a data sample is chosen as test data, the remaining 140 data samples are used as training data. Assuming that a selected test data belongs to the medieval music sub-class in the classical music category, the training data contains 50 classical music, 48 popular music, and 42 rock music data. Based on the above rule, a total of 141 executions were used to build the MGC models.

For each training round, the 140 training data were used to build four MGC models; one for main class classification, and the other three for sub-class classification (i.e., for respective classical, popular and rock). The features significant to the four classes were individually handled by the MCA and FE operations. After this, the kNN and ANN machine learning approaches were used to extract the feature data, including value and weight, to train the MGC models. In experiments, the value of k was 2 for kNN method, and the structure of ANN included a hidden layer with ten neurons. Next, the model testing operation was performed to evaluate the trained models, until the accuracy rate of the models were accepted and thus confirmed. The confirmed main class classification model was able to determine the main class of the musical composition (i.e., classical, popular or rock music). Regardless of model testing or music classification, the confirmed sub-class classification models were able to classify the evaluated data into appropriate sub-classes based on the identified main class.

4.5. Results

The results of different scenarios are listed in Table 4. CI is the case identifier. SC is the class set, which consists of Main, Classical, Popular, Rock, and All-mixed. The All-mixed type is a combination of all the classes, without differentiating between the main class or sub-class. SN represents the number of samples used. LN is the number of the labels applied for indicating the classification output. Taking the Main class as an example: since it is composed of three output labels: Classical, Popular, and Rock, its LN is 3. The remaining LN values of sub-class can be deduced by analogy. MM denotes the adopted machine learning method; either kNN or ANN. FE indicates whether or not the feature extraction is performed. FW indicates whether or not the AHP is carried out to weight the features. TS indicates whether or not the TSC process is performed. Finally, AC is the accuracy rate stating the performance of the MGC for a specific scenario. This study considers the uses of feature weights, feature extraction, and two-step method for comparisons. Table 4 shows that the proposed feature extraction method with AHP feature weights can improve the accuracies of music classification. For instance, the accuracy of music classification based on the proposed method with ANN can be improved to 87.23% for main-class. For two-step classification, the proposed method can extract the important features and give the AHP weights for these features, and the accuracy of the proposed method is 57.19% which is higher than other cases.

Table 4. Results. CI: case identifier; SC: class set; SN: number of samples used; LN: number of the labels applied for indicating the classification output; MM: adopted machine learning method; either artificial neural network (ANN) or k-nearest neighbors (kNN); FE: whether or not the feature extraction is performed; FW: whether or not the AHP is carried out to weight the features; TS: whether or not the TSC process is performed; AC: accuracy rate stating the performance of the MGC for a specific scenario.

CI	SC	SN	LN	MM	FW	FE	TS	AC
1	Main	141	3	kNN	N	Y	N	77.30%
2						N		75.89%
3					Y	Y		80.14%
4						N		85.82%
5				ANN	N	Y		85.11%
6						N		78.72%
7					Y	Y		87.23%
8						N		84.40%

Table 4. Cont.

CI	SC	SN	LN	MM	FW	FE	TS	AC
9	Classical	51		kNN		Y	N	35.29%
10					N	N		45.10%
11						Y		35.29%
12					Y	N		27.45%
13				ANN		Y		52.94%
14					N	N		23.53%
15						Y		78.43%
16					Y	N		49.02%
17	Popular	48	5	kNN		Y	N	14.58%
18					N	N		14.58%
19						Y		14.58%
20					Y	N		22.92%
21				ANN		Y		35.42%
22					N	N		54.17%
23						Y		35.42%
24					Y	N		68.75%
25	Rock	42		kNN		Y	N	30.95%
26					N	N		21.43%
27						Y		30.95%
28					Y	N		26.19%
29				ANN		Y		38.1%
30					N	N		42.86%
31						Y		42.86%
32					Y	N		40.48%
33	All-mixed	141	15	kNN		Y	Y	14.18%
34					N	N		14.89%
35						Y		12.77%
36					Y	N		10.64%
37				ANN		Y		17.73%
38					N	N		13.48%
39						Y		19.15%
40					Y	N		23.40%
41				kNN		Y		24.30%
42					N	N		28.48%
43						Y		21.73%
44					Y	N		27.91%
45				ANN		Y		43.83%
46					N	N		35.10%
47						Y		57.19%
48					Y	N		44.52%

4.6. Discussions

Referring to the review summary by article [3], the proposed TSMGC approach belongs to a machine learning enabled symbolic data-based MGC that uses global-based features. But the TSMGC can enhance the performance of MGC by adding the following points: (1) combines various musical features extracted from the RMH and the calculated entropy, (2) applies the weight of features and their AHP-determined impact values on the basis of probability density function, (3) performs a machine

learning based two-step classification process to more accurately categorize a music into a main-class and further sub-classes.

Based on the experimental results shown in Table 4, this paragraph compares the performance (i.e., accuracy) of the proposed TSMGC method with those of MGC methods whose feature extraction uses musical interval [2], pitch [32] or RMH only. According to Table 5, it implies that TSMGC applying AHP weighted RMH features achieved higher accuracy for each classification model. That is, the proposed TSMGC method considering multiple features delivers better performance than the methods using single features [2,32]. Furthermore, extracting the feature weights by AHP improves the MGC performance so that it is better than directly using RMH without weights.

Table 5. Comparison of classification accuracy of different feature extraction methods. RMH: rhythm, melody, and harmony; AHP: analytic hierarchy process.

Feature Extraction Method	Number of Feature Values	Main-Class	Subclass of Classical Music	Subclass of Popular Music	Subclass of Rock Music
Musical interval only	12	59.57%	21.57%	22.92%	19.50%
Pitch only	50	74.47%	25.49%	20.83%	19.05%
RMH	104	85.11%	52.94%	35.42%	38.10%
RMH and AHP	104	87.23%	78.43%	35.42%	42.86%

In addition, since the TSMGC is a two-step classification (TSC) method capable of precisely categorizing the result into not only a main class but also corresponding sub-classes, it is worth evaluating if TSC provides better performance than one-step classification (OSC) methods [2,32]. In theory, the OSC method trains a single classification model and completes the whole classification operation in one step, while TSC first determines one model for main classes, and several models for sub-classes. Table 6 presents the accuracy comparisons between OSC and TSC. It shows that the accuracy rate of TSC is higher than that of OSC for all FE methods. In particular, TSC in conjunction with RMH and an AHP based FE method delivers an accuracy rate of 57%, which is significantly higher than that of 19% produced by OSC, even with RMH and AHP supported features.

Table 6. Comparison of classification accuracy of one-step and two-step methods.

Feature Extraction Method	One-Step Classification	Two-Step Classification
Musical interval only	7%	24%
Pitch only	15%	20%
RMH	18%	44%
RMH and AHP	19%	57%

5. Conclusions and Future Work

This study proposes a two-step MGC approach based on AHP weighted musical features, called TSMGC. The TSMGC approach adopts a FE method to analyze the distribution of the values of each of the RMH features and to calculate the intersection area of distributions in varied classes in order to extract significant features. Additionally, AHP was applied to adjust the weight of each extracted feature in order to improve MGC performance. Moreover, the TSC algorithm helps to determine the main class and the corresponding sub-class of the target music. Experimental results prove that TSMGC performance is superior to those of traditional MGC methods.

Although the overall accuracy rate of TSMGC is 1 to 2 times better than that of traditional MGC methods, the accuracy rate of its sub-class classification is still lower than 50% due to the similarity of features among the classes. The extraction of more musical features like timbre would address this problem. The idea is that, since a music composition can present different timbres according to different spectrums and waveforms, it can cause different stimuli to human senses. So that, humans can distinguish arpeggios performed by different people, as well as sounds produced by different musical

instruments. In reality, each music genre has its frequently used instrument combinations or has some specific singing voices. Therefore, including timbre as a feature in future investigations is expected to improve MGC accuracy. In addition, the sample size of datasets and the number of music genres can be extended for TSMGC enhancement in the future. Furthermore, some techniques (e.g., restricted Boltzmann machine, deep belief nets, deep Boltzmann machine, ensemble techniques, cross-validation, etc.) can be applied for the avoidance of overfitting issues for further TSMGC extended investigations.

With the rapid growth of music applications, effective and efficient automatic MGC mechanisms are becoming increasingly important for novel music applications, such as online music platforms. The proposed TSMGC composed of a RMH multi-feature-based MCR and MCA, an AHP-supported FE, as well as an ML-based TSC, is capable of performing MGC efficiently and accurately. TSMGC addresses the important research direction of extracting AHP weighted features for use in the MGC process. The experiment conducted in this study also provides a valuable and practical reference for designing and implementing new systems for automatic MGC applications.

Author Contributions: Y.-T.C. and C.-H.C. conceived and designed the experiments; S.W. performed the experiments; Y.-T.C., C.-H.C. and C.-C.L. analyzed the data; S.W. contributed reagents/materials/analysis tools; Y.-T.C., C.-H.C. and S.W. wrote the paper.

Funding: This work was supported by the Ministry of Science and Technology, Taiwan [grant number 106-2221-E-239-003]. This work was also supported by Fuzhou University, China [grant number 510730/XRC-18075].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lee, J.H.; Downie, J.S. Survey of music information needs, uses, and seeking behaviours: Preliminary findings. In Proceedings of the 5th International Symposium on Music Information Retrieval, Barcelona, Spain, 10–14 October 2004; pp. 1–4.
2. Lo, C.C.; Kuo, T.H.; Kung, H.Y.; Chen, C.H.; Lin, C.H. Personalization music therapy service recommendation system using information retrieval technology. *J. Inf. Manag.* **2014**, *21*, 1–24.
3. Corrêa, D.C.; Rodrigues, F.A. A survey on symbolic data-based music genre classification. *Expert Syst. Appl.* **2016**, *60*, 190–210. [\[CrossRef\]](#)
4. Conklin, D. Multiple viewpoint systems for music classification. *J. New Music Res.* **2014**, *42*, 19–26. [\[CrossRef\]](#)
5. Zhong, J.; Cheng, Y.F. Research on music mood classification integrating audio and lyrics. *Comput. Eng.* **2012**, *38*, 144–146.
6. Hu, X.; Downie, J.S.; Ehmann, A.F. Lyric text mining in music mood classification. In Proceedings of the 10th International Society for Music Information Retrieval Conference, Kobe, Japan, 26–30 October 2009.
7. Zeng, Z.; Pantic, M.; Roisman, G.I.; Huang, T.S. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Trans. Pattern Anal. Ma-Chine Intell.* **2009**, *31*, 39–58. [\[CrossRef\]](#)
8. Kermanidis, K.L.; Karydis, I.; Koursoumis, A.; Talvis, K. Combining language modeling and LSA on Greek song “words” for mood classification. *Int. J. Artif. Intell. Tools* **2014**, *23*, 17. [\[CrossRef\]](#)
9. Silla, C.N., Jr.; Freitas, A.A. Novel top-down approaches for hierarchical classification and their application to automatic music genre classification. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, USA, 11–14 October 2009; pp. 3499–3504.
10. Scaringella, N.; Zoia, G.; Mlynek, D. Automatic genre classification of music content—A survey. *IEEE Signal Process. Mag.* **2006**, *23*, 133–141. [\[CrossRef\]](#)
11. Fu, Z.; Lu, G.; Ting, K.M.; Zhang, D. A survey of audio-based music classification and annotation. *IEEE Trans. Mult.* **2011**, *13*, 303–319. [\[CrossRef\]](#)
12. Duda, R.O.; Hart, P.E.; Stork, D.G. *Pattern Classification*; John Wiley & Sons Inc.: New York, NY, USA, 2001.
13. Da Fontoura Costa, L.; César, R.M., Jr. *Shape Analysis and Classification*; CRC Press: Boca Raton, FL, USA, 2001.
14. Lykartsis, A.; Wu, C.W.; Lerch, A. Beat histogram features from NMF-based novelty functions for music classification. In Proceedings of the International Conference on Music Information Retrieval, Malaga, Spain, 26–30 October 2015.
15. Lin, C.R.; Liu, N.H.; Wu, Y.H.; Chen, A.L.P. Music classification using significant repeating patterns. *Lect. Notes Comput. Sci.* **2004**, *2973*, 506–518.

16. Chew, E.; Volk, A.; Lee, C.Y. Dance music classification using inner metric analysis. *Oper. Res./Comput. Sci. Interfaces Ser.* **2005**, *29*, 355–370.
17. Thomas, L.S. Music helps heal mind, body, and spirit. *Nurs. Crit. Care* **2014**, *9*, 28–31. [[CrossRef](#)]
18. Drossinou-Korea, M.; Fragkouli, A. Emotional readiness and music therapeutic activities. *J. Res. Spec. Educ. Needs* **2016**, *16*, 440–444. [[CrossRef](#)]
19. Karydis, I. Symbolic music genre classification based on note pitch and duration. In *Advances in Databases and Information Systems (ADBIS 2006)*; Lecture Notes on Computer Science; Manolopoulos, Y., Pokorný, J., Sellis, T.K., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4152, pp. 329–338.
20. Saaty, T.L. A scaling method for priorities in hierarchical structures. *J. Math. Psychol.* **1977**, *15*, 234–281. [[CrossRef](#)]
21. Pincus, S.M.; Gladstone, I.M.; Ehrenkranz, R.A. A regularity statistic for medical data analysis. *J. Clin. Monit. Comput.* **1991**, *7*, 335–345. [[CrossRef](#)]
22. Richman, J.S.; Moorman, J.R. Physiological time-series analysis using approximate entropy and sample entropy. *Am. J. Physiol. Heart Circ. Physiol.* **2000**, *278*, H2039–H2049. [[CrossRef](#)] [[PubMed](#)]
23. Bishop, C. *Pattern Recognition and Machine Learning*; Springer: Berlin, Germany, 2006; ISBN 0-387-31073-8.
24. Saaty, T.L. How to make a decision: The analytic hierarchy process. *Eur. J. Oper. Res.* **1990**, *48*, 9–26. [[CrossRef](#)]
25. Amancio, D.R.; Comin, C.H.; Casanova, D.; Travieso, G.; Bruno, O.M.; Rodrigues, F.A.; Costa, L.D.F. A systematic comparison of supervised classifiers. *PLoS ONE* **2014**, *9*, e94137. [[CrossRef](#)]
26. Mandel, M.I.; Ellis, D.P.W. Song-level features and support vector machines for music classification. In *Proceedings of the 6th International Conference on Music Information Retrieval: ISMIR-05*, London, UK, 11–15 September 2005; pp. 594–599.
27. Lee, H.; Largman, Y.; Pham, P.; Ng, A.Y. Unsupervised feature learning for audio classification using convolutional deep belief networks. *Adv. Neural Inf. Process. Syst.* **2009**, *22*, 1096–1104.
28. Brown, A.R. *Making Music with Java*; Lulu Press, Inc.: Morrisville, NC, USA, 2009.
29. Aruffo, C.; Goldstone, R.L.; Earn, D.J.D. Absolute judgment of musical interval width. *Music Percept. Interdiscip. J.* **2014**, *32*, 186–200. [[CrossRef](#)]
30. Han, J.; Kamber, M.; Pei, J. *Data Mining: Concepts and Techniques*, 3rd ed.; Morgan Kaufmann Publishers: Chusetts, MA, USA, 2011.
31. Chen, C.H.; Lin, H.F.; Chang, H.C.; Ho, P.H.; Lo, C.C. An analytical framework of a deployment strategy for cloud computing services: A case study of academic websites. *Math. Probl. Eng.* **2013**, *2013*, 14. [[CrossRef](#)]
32. Shan, M.K.; Kuo, F.F. Music style mining and classification by melody. *IEICE Trans. Inf. Syst.* **2003**, *E86-D*, 655–659.

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).