Evolutionary Approximation of Instrumental Texture in Polyphonic Audio Recordings

Igor Vatolkin
Department of Computer Science
TU Dortmund
Dortmund, Germany
0000-0002-9454-9402

***Abstract*—We propose a novel approach to extract audio features based on evolutionary approximation of instrumental texture in polyphonic audio recordings. A population of mixtures of samples from 51 instruments with 165 individual instrument bodies or playing styles is evolved with the help of musically meaningful genetic operators to produce chords which are as similar as possible to unknown signals. Our algorithm allows for a simultaneous approximation of all onsets/chords from a given audio track. The fitness function is designed to retain mixtures which are not directly comparable because they approximate different segments of a track like intro or verse. Another advantage is that no labelled signals are required to learn supervised models for instrument prediction, and the sample database can be easily extended with further instruments. Although the multi-label classification performance of instrument recognition still has room to be improved, the derived instrumental and pitch statistics are comparable to the best selected semantic features from a large set of 566 descriptors including not only instrument and pitch statistics, but also chord, harmony, structure, temporal, dynamics, emotional, vocal, and further characteristics, even outperforming them for a half of tested music categories.**

***Index Terms*—Evolutionary music approximation, instrument recognition, genre recognition, semantic features**

I. Introduction and Related Work

Instrument recognition in polyphonic audio is a very challenging classification task in music data analysis. In contrast to the recognition of individual samples in earlier works [1], [2], simultaneously played instruments contribute to the audio signal with very different individual properties of distributions of overtones and non-harmonic frequencies, temporal change of sound during attack, decay, sustain, and release tone phases [3], playing style, or applied effects like reverb or distortion. More recent works introduced enhanced techniques, like engineering of specific domain features [4], optimisation of classification models with feature selection [5], source signal separation [6], or deep neural networks [7], [8].

A common approach to recognise instruments starts with the audio time signal and tries to identify relevant properties of given instruments after various transforms, like the estimation of spectrum or cepstrum. However, an oppositely directed procedure based on synthesis can be also considered, namely the combination of individual tone samples to a polyphonic mixture which should approximate an unknown chord. As the number of possible combinations of instruments and individual bodies, pitches, playing styles, loudness levels, and applied effects can be theoretically infinite (e.g., if loudness is measured by continuous values), evolutionary algorithms (EAs) can be considered as a method to explore such a huge search space trying to identify polyphonic mixtures which will approximate the unknown sound as perfect as possible.

There exist many systems which applied EAs for music composition, e.g., Jazz solo generation [9], adjustment of parameters of granular synthesis [10], or generation of background music [11]. A good overview of earlier applications is provided in [12]. However, to our knowledge, in none of the published studies EAs were applied to approximate "real-world" classical or popular polyphonic music pieces with the target to identify underlying instrumental and pitch texture.

In this work, we introduce a method how this can be implemented with an EA. However, it is important to mention that such optimisation task can not always be solved, not only because there exist different instruments with similar sounds (like violin and viola), but also because the boundaries between instruments may vanish when instrument samples are synthesised by software (as in digital pianos) or strongly augmented during production in studios.

That's why the primary goal of our algorithm is not to achieve the perfect performance in instrument recognition, but to derive mid-level relevant properties of instrumental texture in audio tracks which may improve further related applications: genre recognition, personal music recommendation, audio structure analysis, etc. Thus, the features extracted after our evolutionary approximation should not be directly treated as exactly identified instruments, but rather as similarities to concrete instrument samples in our database which nevertheless makes them semantically and musically meaningful.

In Section II, we describe the operating principle of a proposed EA to approximate polyphonic audio recordings. To get a better insight into the complexity of instrument recognition task, in the first study we measure the quality of multi-label instrument recognition in artificially generated music pieces, see Section III. In Section IV, we describe how instrumental and pitch properties can be extracted and applied to genre recognition and evaluate them for the prediction of 14 music genres and styles. The classification performance is also compared to a simple baseline and an extensive set of various high-level semantic descriptors. Section V provides a summary of results and ideas for future work.

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

II. Algorithm

A. Sample Database

Our database of instrument samples [13] is compiled from several sources: the instrument sample database from [5], Ethno World 5 Professional and Voices samples [14], and Komplete 11 Ultimate samples [15]. Because many of 51 instruments are represented with different individual instrument bodies and playing styles (e.g., Alicia's Keys, The Gentleman, The Giant, The Grandeur, The Maverick pianos), the overall number of sample categories is 165. Table I provides an overview of all instruments with numbers of corresponding styles and sources.

B. Operating Principle and Parameters

Obviously, in some meaningful feature space the distances between feature vectors which represent same or similar instruments should be smaller than distances between feature vectors which represent very different instruments. An example is shown in Figure 1. The feature domain is built with the chromagram or pitch class profile [16] which measures the strengths of halftones. Subfigure (b) shows the chromagram around the approximated chord from the original recording which is marked with a red rectangle in the score, Subfigure (a). Subfigure (c) shows the chromagram for the mixture of instrument samples which are also present in the original chord, (d) for the mixture with one tone removed, (e) for the mixture with two pitches shifted, and (f) for the mixture with one sample played by piano and not violin. As a distance measure, we calculated the average absolute difference between all chroma values for each time window of the corresponding plot matrices. Subfigure (g) illustrates that the approximation with correct instruments and pitches has the smallest distance to the original chord for time windows 1 and 4-9. Note that the tones of the mixture come from our database and belong to other instrument bodies than in the original audio recording.

Now consider that we have some mixture of tones which should approximate an unknown chord. The distance to audio features of this chord can act as a fitness function which evaluates the quality of the given mixture. With the help of evolutionary operators, the mixture can be changed: e.g., a new tone may be added or an existing one removed, a pitch of a particular tone can be shifted, or an instrument can be replaced by another one. For instance, a mutation which adds a tone could produce approximation 1, Subfigure (c), from approximation 2, Subfigure (d), where the approximation 1 has a smaller distance to the approximated chord in the chromagram feature domain.

The evolutionary algorithm for the approximation of polyphonic recordings operates as follows. During the initialisation stage, μ individuals (mixtures of one to five instrument tone samples) are created. The instrument, the style, and the pitch are drawn randomly. The probability to use only one sample is set to 10%, two samples to 30%, three samples to 30%, four samples to 20%, and five samples to 10%. In each iteration step, λ offsprings are generated from randomly selected parent solutions. In the final experiments for this study, the best results were achieved with $\mu = 400$ and $\lambda = 1$. These and also later mentioned parameter values were carefully selected based on the first experiments and must not be the optimal; an exhaustive evaluation of many possible settings was beyond the scope of this study but should be addressed in future.

Three different mutation operators are currently implemented. The first one changes the number of mixed samples. The probability to increase the number of samples $P(m_1)$ changes from 100% to 80%, 40%, 10%, and 0% for mixtures of 1, 2, 3, 4, and 5 samples, respectively. If the drawn random number is below $P(m_1)$, then exactly one randomly selected sample is added to the mixture. If this number is equal to or above $P(m_1)$, then a randomly selected sample is removed from the mixture. The second mutation shifts the pitch of a random sample, adding $\lfloor 15 \cdot G \rfloor$ to the current pitch, where G is a Gaussian number with mean 0 and standard deviation 1. In case the new pitch is below the lowest or above the highest possible pitch of the instrument of that sample, the new pitch is set to the corresponding boundary. The third mutation exchanges an instrument for a randomly drawn sample, keeping its pitch, and selecting a new style by chance. For our initial study, we have restricted the genetic operators to these three, but plan to implement more operators in further experiments, like loudness change mutation or crossover.

For each offspring, the number of applied mutations is set to $\lfloor G \cdot \alpha + \beta \rfloor$, with a restriction that at least l_{bound} and at maximum u_{bound} mutations are applied. After the initial experiments, we found the settings $\alpha = 6$, $\beta = 3$, $l_{bound} = 1$, and $u_{bound} = 10$ to perform quite well. For instance, the application of exactly one mutation for each offspring reduced the scale of search space exploration and led to worse performance.

To approximate a complete audio track with all notes/chords for which no exact score or symbolic representation like MIDI is available, at first all onset events must be extracted (i.e., all time positions where at least one new tone begins). For the fitness evaluation, we measure the absolute distance between normalised feature values of the approximated onset and a candidate mixture. The estimation of feature vectors is based on two parameters: a feature domain and a feature processing method. Among several examined feature domains, the Mel spectrum was found to be the significantly best method (see Section III for details). For feature processing, we estimated the attack phase with librosa [17] (the time interval between the beginning of the new tone or chord and the time point with the next energy peak). In the "complete" processing, the distance is measured across all feature values from time frames between the approximated onset and the the next onset. For the final experiments on genre recognition, we also tested the "attack-release" ("AR") processing, where only the feature values from the middle of the attack phase, the end of the attack phase, and from the middle of the release phase are stored. This is motivated by the assumption that non-harmonic frequencies like piano key stroke during the attack phase may be useful to identify instruments and should not be mixed with

TABLE I
LIST OF INSTRUMENT SAMPLES USED IN THE STUDY. COLUMN "NO." LISTS THE NUMBER OF STYLES.

Name	No.	Source	Name	No.	Source	Name	No.	Source
Acoustic guitar	12	[5]	Drums	12	[15]	Panflute	1	[14]
Balalaika	1	[14]	Dung dkar trumpet	1	[14]	Piano	11	[5], [15]
Bandura	1	[14]	Egyptian fiddle	1	[14]	Pinkillo	1	[14]
Banjo framus	1	[14]	Electric bass	7	[15]	Pivana flute	1	[14]
Banjolin	1	[14]	Electric guitar	9	[5]	Saxophone	5	[15]
Bass	2	[15]	Electric piano	6	[15]	Scale changer harmonium	1	[14]
Bassoon	2	[15]	Erhu	1	[14]	Shakuhachi	1	[14]
Bawu	1	[14]	Flute	8	[5], [15]	Sitar	1	[14]
Bouzouki	1	[14]	Fujara	1	[14]	Tampura	1	[14]
Cello	9	[5], [15]	Horn	4	[15]	Tanbur	1	[14]
Ceylon guitar	1	[14]	Jinghu opera violin	1	[14]	Trombone	4	[15]
Clarinet	2	[15]	Kantele	1	[14]	Trumpet	11	[5], [15]
Contrabassoon	1	[15]	Melodica	1	[14]	Tuba	3	[15]
Cumbus	1	[14]	Morin khuur violin	1	[14]	Turkey saz	1	[14]
Dallape accordion	1	[14]	Oboe	2	[15]	Ukulele	1	[14]
Dilruba	1	[14]	Oud	1	[14]	Viola	9	[5], [15]
Domra	1	[14]	Organ	6	[15]	Violin	10	[5], [15]

a more stable and harmonic sound after the end of the attack phase (cf. [18]). We have also conducted further experiments storing values from the onset frame, the end of the attack phase and the last frame of the release phase, however, the results were not better or even worse.

A problematic issue is that the evolutionary approximation of each onset in a music track may be very time consuming, because a typical popular music piece may contain more than thousand onsets. However, we can assume that many of them have the same or closely related instrument tones, because chord, harmonic, and instrumental properties repeat within an individual music piece. Therefore, our approach approximates all onsets simultaneously. For each candidate mixture, we measure distances to all onsets in the music piece. Then, we sort these distances in ascending order and estimate the mean value of ϕ per cent of the smallest distances between a candidate mixture and the best approximated onsets. The value of ϕ should be chosen carefully. Setting it to a too low value will prioritise solutions which are very specific and approximate at best a sole onset or a couple of similar onsets. Setting it to a too high value will prioritise very general solutions which approximate a large number of onsets and the general structure of the music piece, but are not very precise to recognise individual onsets. Assuming that the shortest segments of typical popular tracks like intro or bridge may continue for around 5-10 per cent of track length, we have experienced with different ϕ values and set them to 5% and 10% for the final study.

An illustration to fitness estimation is provided in Figure 2. The horizontal axis corresponds to 100 of 520 best approximated onsets in Vivaldi's *The Spring - I. Allegro*, sorted by ascending order with respect to the mean absolute distance between the Mel spectrograms of the approximation and onsets (vertical axis). The height of dark grey shaded rectangles is equal to the mean distance to 5% (26) of the best approximated onsets and the height of light grey shaded rectangles to the mean distance to 10% (52) of the best approximated onsets. This mean distance is used as a fitness function for evolutionary selection. As we observe, the first approximation with correct instruments and pitches (the left subfigure) has smaller (better) fitness values than the approximation with less tones (the middle subfigure) or with shifted tones (the right subfigure). Again, it is important to mention that the instrument bodies and playing styles in the approximations are not the same than in the original recording of the music track.

It can be also thinkable to keep mixtures in the population which approximate well a larger number of onsets in a music track. Therefore, we extended our method to support several subpopulations with different evaluation scopes, some of them targeted to be rather specific (with a smaller ϕ) and some rather generic (with a larger ϕ). We have tested the following implementation. For each of μ initial mixtures, we decide with the equal probability whether it should be a "specialist" ($\phi \in [1, ..., 5], \phi \in \mathbb{N}$), "allrounder" ($\phi \in [6, ..., 20]$), or "generalist" ($\phi \in [21, ..., 100]$). The value of ϕ is randomly drawn from the corresponding intervals and is kept for each individual during the complete optimisation process. It is also inherited to its offspring, so that the sizes of all subpopulations remain the same. During the selection of a new generation, we compare the fitness values of offsprings to fitness values of their subpopulations only, so that a better fitness of a "specialist" mixture which very well optimises some particular onset but poorly approximates almost all other onsets should not replace an "allrounder" mixture with a larger fitness but other optimisation scope. In the following, we mark the setting for variable fitness estimation with subpopulations with $\phi = \text{VAR}$.

After the evaluation of offsprings, μ individuals with the smallest fitness values are selected for the next generation. The evolutionary loop continues for a given number of generations g. In the final study, $g = 3000$ was found to be a good compromise between the runtime and convergence behaviour.

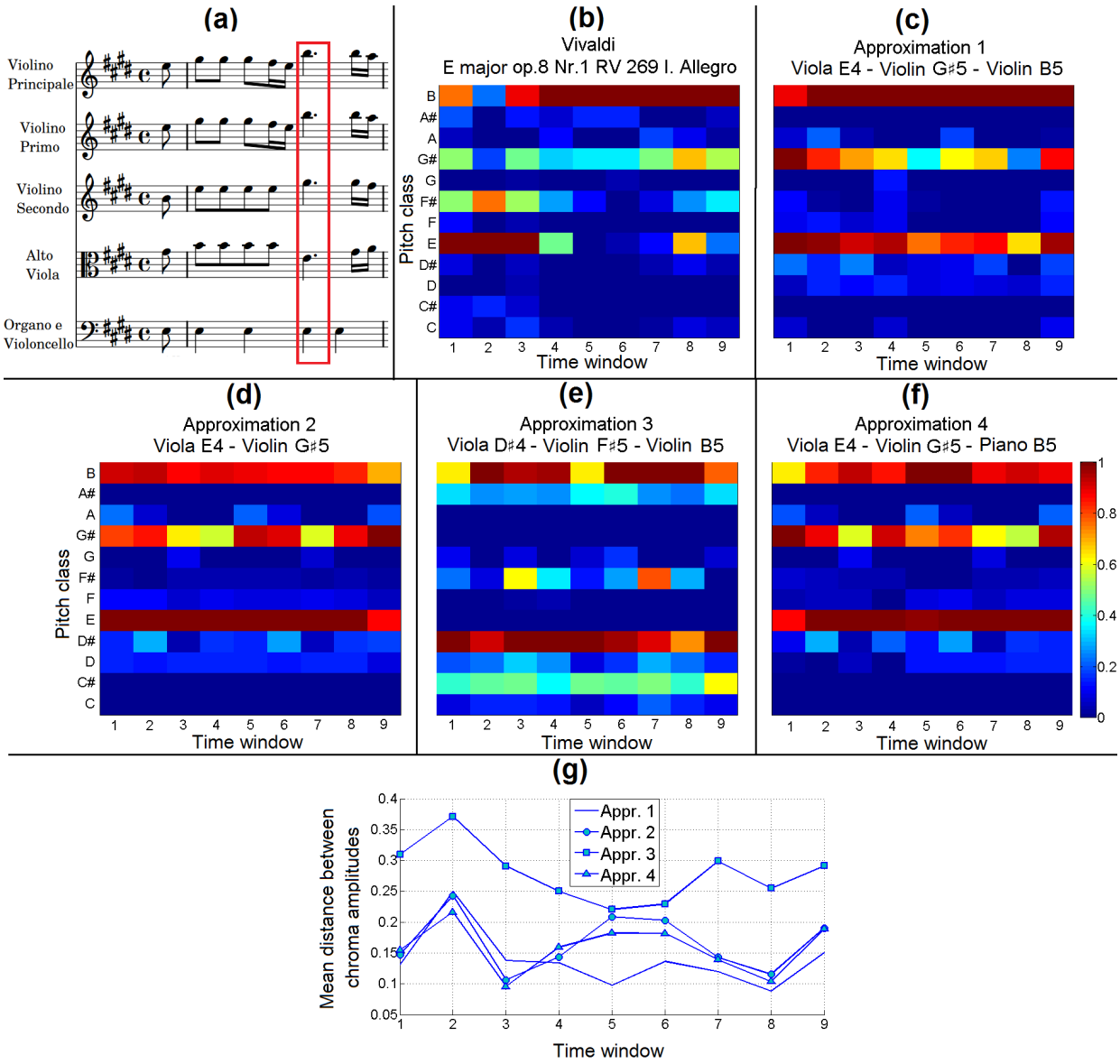

Fig. 1. Approximation of a chord in Vivaldi's *The Spring - I. Allegro*. (a): The score (the chord to approximate is marked with a rectangle); (b): chroma domain from the original recording; (c)-(f): chromagrams of several approximations; (g): distances between approximations and the original recording in chroma domain.

III. Experiment 1: Selection of Feature Domain for Instrument Recognition

A. Setup

To measure the instrument recognition performance of our method, we need polyphonic music pieces with exact annotations of played instruments. For that sake, we created an artificial database of 8 tracks with two simultaneously playing istruments with the help of JFugue [19], one half of tracks with additional drums and another one without. The generated MIDIs were transformed to audio with Kontakt Player using Komplete 11 samples with smooth transitions between onsets and a more natural sound compared to a simple assignment of individual instrument samples to MIDI events.

We tested four feature domains for the estimation of distance between approximations and onsets. Mel frequency cepstral coefficients (MFCCs) [20] (13 dimensions, jAudio implementation [21]) and the complete Mel spectrum (128 dimensions, librosa implementation [17]) are based on the cepstrum (a product of the inverse Fourier transform applied to the logarithm of the squared spectrum [22]) which is further adjusted to the Mel scale, so that the distances between pitches should be perceived as similar by human listeners [23]. The chromagram (12 dimensions, Yale implementation [24]) and the semitone spectrum (85 dimensions, NNLS Chroma implementation [25]) measure the strengths of individual halftones.

After some preliminary trials, we tested two different mutation strengths: weak ($\alpha = \beta = l_{bound} = u_{bound} = 1$) and

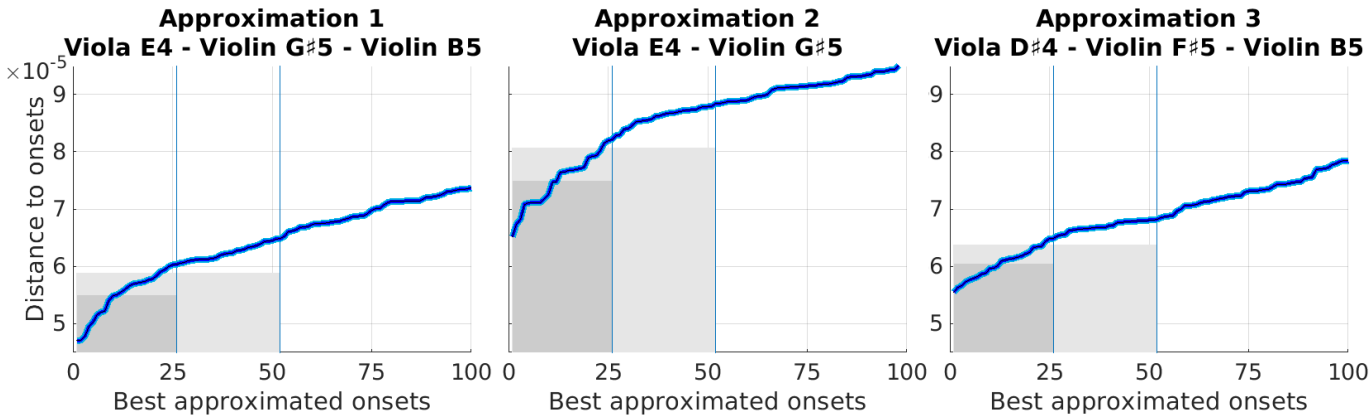

Fig. 2. Example for the fitness evaluation of three approximations applied to a complete Vivaldi's *The Spring - I. Allegro*. For details, see the text.

strong ($\alpha = 6$, $\beta = 3$, $l_{bound} = 1$, $u_{bound} = 10$).

As an evaluation measure (error e), we calculated the share of present instruments which are not contained in the best approximation for each onset. As discussed before, multi-label recognition of instruments is very challenging because of the large number of instruments and also different instruments with similar sounds. To illustrate this, let us estimate the error of a random classifier for instrument recognition in a mixture of two samples from two different instruments for our experiment setup. Among all combinations of available instruments, the number of possible candidate mixtures with two different instruments and $e = 1$ (no instrument was correctly predicted) is equal to $\frac{49 \cdot 48}{2} = 1176$ (the complete number of available instruments is 51, two of them are already in the mixture to approximate). The number of possible approximations with exactly one correctly predicted instrument ($e = 0.5$) is equal to $49 + 49 = 98$ (for each of two present instruments, 49 other instruments remain in the database). The number of possible approximations with $e = 0$ is equal to 1. Then, the overall number of possible candidate mixtures is $1176 + 98 + 1 = 1275 = \frac{51 \cdot 50}{2}$. Under a simplified assumption that all combinations of two instrument samples have equal probabilities to appear in a candidate mixture, the mean expected error of a random classifier would be equal to $\frac{1176 \cdot 1 + 98 \cdot 0.5 + 1 \cdot 0}{1275} = 0.96078$.

After the similar calculation, the error of a random classifier which approximates one sample with another one is $e = 0.98039$ and three samples with a mixture of exactly three samples is $e = 0.94084$; note that the exact theoretic analysis is very complex because, e.g., two sample mixtures can be approximated by three sample mixtures, etc., and the numbers of samples in the mixtures after the optimisation depend on the parameters of an EA and on a concrete classification task.

B. Results

Table II presents the errors. The instrument recognition performance of the Mel spectrogram significantly outperforms all three other feature domains, as confirmed by the Wilcoxon signed rank test applied for errors from different music tracks.

TABLE II
PERFORMANCE OF MULTI-LABEL INSTRUMENT RECOGNITION FOR ARTIFICIAL POLYPHONIC TRACKS. THE SMALLEST ERROR IS MARKED WITH THE BOLD FONT.

Mutation strength	Chromagram	Semitone spectrum	MFCCs	Mel spectrum
weak	0.8942	0.8631	0.8671	0.7845
strong	0.8990	0.8474	0.8359	**0.7558**

The stronger mutation led also to smaller errors, however the difference was not significant. Although the best achieved error of 0.7558 is rather high, it is still significantly lower than the error of a random classifier (see the example above). The main challenge is here that samples of different instruments are sometimes very similar (this depends also on the pitch). For instance, in one observed case, a replacement of a violin sample with a piano sample for a string recording led to a small decrease of distance between the candidate mixture and the onset to approximate.

However, a large difference of errors between the worst value with the chromagram ($e = 0.8990$) and the best value with the Mel spectrum ($e = 0.7558$) points out a high potential for a further optimisation of feature domain, i.e. the identification of the most representative and distinctive features.

IV. Experiment 2: Genre Recognition

A. Setup

For genre recognition, we follow the setup of [26], where 6 music genres and 8 styles are predicted, and only small training sets of 20 tracks are used, so that this situation very well matches a real-world scenario when a user wants to spend less efforts to define a rather small training set of some personal category. For the identification of the smallest feature sets with smallest errors by means of evolutionary multi-objective feature selection, other 120 tracks (optimisation set OS120 [27] are used. The final validation is done on the album-independent test set TAS120 with 120 tracks which

had the same genre distribution as OS120 tracks, but are represented with other artists and albums.

With regard to the results from Section III, we estimated the fitness measure with respect to distances in the Mel spectrum and used a stronger mutation for all experiments on genre recognition. Two different feature processing methods ("complete" and "AR"), as well as three $\phi \in \{5, 10, \text{VAR}\}$ values were used, see Section II.

To focus our study more on different statistics of evolutionary approximations of music pieces, we restricted the classification algorithm to only random forest [28] with 100 trees, because this method is very robust, fast, has only few parameters to setup, and is capable to deal with very small training sets, in contrast to deep neural networks, which typically require larger training sets and are not so optimal for our application scenario. As in [26], the classification windows are 4s with 2s overlap, and the music tracks are assigned to categories by majority voting.

The following statistics are proposed as approximative features. For each instrument and each onset in the approximated music track, we save the smallest distance between the best candidate mixture which contains this instrument and the onset to approximate. These smallest distances are kept during the complete evolutionary loop in an archive and do not represent the final population only. Then, we estimate the mean, the minimum, and the maximum values for each of 51 instrument and 88 theoretically possible pitches for two different analysis frames of 10s and 3s. Additionally, we sort the recognised instruments based on the smallest distances, and assign ranks to corresponding approximative features, e.g., value of "rang of acoustic guitar" = 1 means that acoustic guitar had the smallest mean distance between approximations with this instrument and unknown onsets in the analysis frame. This leads to an overall number of feature dimensions equal to $(51 \cdot 3 + 88 \cdot 3 + 51) \cdot 2 = 936$[1].

We compare the approximative features to semantic features from the previous work [29] which are summarised in Table III. Note that this baseline set contains a large number of very different music properties. Further details and references to individual features are provided in [26].

Because the distribution of genres in optimisation and test sets was not balanced, the evaluation measure is the balanced relative error:

$$m_{BRE} = \frac{1}{2}\left(\frac{FN}{TP+FN} + \frac{FP}{TN+FP}\right), \qquad (1)$$

with TP denoting the number of true positives (classification instances which are correctly predicted as belonging to the positive category in supervised binary classification), TN the true negatives (correctly predicted negative instances), FP the false positives (wrongly predicted negative instances), and FN the false negatives (wrongly predicted positive instances).

[1]Please note that this calculation is done for simplicity reasons; because many instruments have a smaller range of pitches compared to piano, the number of meaningful dimensions is less than 936.

B. Results

Table IV presents the summary of classification errors for 6 genres (Classic–R'n'B) and 8 styles (AdultContemporary–Urban). As a simple baseline, the second column ("Mel spect.") contains m_{BRE} values for the classification with random forest using the Mel spectrum (recall that this feature domain is also used for the estimation of fitness values for evolutionary approximations of audio tracks). The second baseline ("Semantic", the third column) contains values from [26] achieved with the best selected semantic features from a large set with very different interpretable harmonic, instrumental, temporal, melodic, emotional, and other properties. As it can be expected, these features are significantly better than the Mel spectrum baseline (p = $1.2207\text{e}{-4}$, the Wilcoxon signed rank test for the comparison of values across the categories, with the default level of significance of 5%).

In the 4th column ("All"), the errors are reported for all estimated approximative features, as described in the previous subsection. The corresponding best configuration of a feature processing method and a fitness evaluation method is provided in the 5th column. Although m_{BRE} is better (smaller) than for the Mel spectrum baseline for 9 of 14 categories, the difference is not significant (p = 0.1573), and these features are worse than the semantic baseline (p = $1.2207\text{e}{-4}$).

However, the situation changes after the application of evolutionary multi-objective feature selection[2] with the goal to identify the smallest subsets with as low as possible errors among all approximative features. To avoid overfitting, the models are always created from training sets, feature selection is validated on the optimisation set OS120, and the best errors on the independent test TAS120 are stated in the 6th column of the table. Again, the corresponding configuration of feature processing and fitness evaluation strategy is provided in the table (7th column).

First of all, feature selection leads to significantly smaller test errors compared to all approximative features (p = 0.0031). Only for the category SoftRock, the test error increases after the feature selection which means that the best features identified for the optimisation set have a poor generalisation ability (overfitting effect). Second, the best selected approximative features outperform the 1st baseline (Mel spectrum) in all cases (p = $1.2207\text{e}{-4}$). Third, the classification performance of the best approximative features is not significantly different from the second baseline (p = 0.2958), and the best selected approximative features are better than the best selected semantic features for a half of categories: Classic, Electronic, Jazz, AlbumRock, HeavyMetal, ProgRock, and Urban.

Such a good performance is particularly surprising because our approximative features describe only instrumental and pitch properties, in contrast to the large semantic feature set with very different musically meaningful characteristics partly based on previously trained and optimised supervised

[2]We applied $\mathcal{S}$-metric selection evolutionary multi-objective algorithm (SMS-EMOA) [30] to minimise m_{BRE} and the number of selected features; for further details and parameters we refer to [26].

TABLE III
SEMANTIC BASELINE FEATURES FROM THE PREVIOUS WORK [26] (DIM.: NUMBER OF DIMENSIONS).

Group	Examples	Dim.
Chord statistics	Number of different chords and chord changes in 10s, shares of the most frequent chords	5
Chroma and harmony	Consonance, key, strengths of pitch intervals, tonal centroid	258
Instruments	Share of guitar, piano, strings, wind instruments in 10s	32
Moods	Aggressive, earnest, energetic, sentimental	64
Structural complexity	Complexity of chords, harmony, instruments	70
Tempo, rhythm, and structure	Beats per minute, duration of a music piece, rhythmic clarity	9
Various features	Activation level, vocal descriptors, characteristics of melodic range	128

TABLE IV
BALANCED CLASSIFICATION ERRORS FOR GENRE AND STYLE RECOGNITION. BASELINES: MEL. SPECT.: CLASSIFICATION WITH THE RANDOM FOREST USING THE MEL SPECTRUM (128 DIMENSIONS); SEMANTIC: BEST MODELS FROM THE FOUR CLASSIFICATION METHODS AND A LARGE SET OF HIGH-LEVEL SEMANTIC DESCRIPTORS AFTER [26] (566 DIMENSIONS). APPROXIMATIVE FEATURES: ALL: ALL APPROXIMATIVE FEATURES (936 DIMENSIONS, ESTIMATED SEPARATELY FOR 6 COMBINATIONS OF FEATURE PROCESSING AND FITNESS EVALUATION METHODS); BEST: BEST APPROXIMATIVE FEATURES FOUND AFTER FEATURE SELECTION (VARIABLE NUMBER OF DIMENSIONS). FEATURE PROCESSING METHODS: A: "AR"; C: "COMPLETE". THE SMALLEST m_{BRE} VALUES FOR EACH CATEGORY ARE MARKED WITH THE BOLD FONT.

Category	Baselines		Approximative features			
	Mel spect.	Semantic	All	FP, ϕ	Best	FP, ϕ
Classic	0.0286	0.0276	0.0619	A,VAR	**0.0238**	C,10
Electronic	0.2238	0.1610	0.2762	A,5	**0.1476**	C,VAR
Jazz	0.1952	0.1400	0.1810	C,10	**0.1286**	C,VAR
PopRock	0.4778	**0.1575**	0.4200	A,10	0.3178	C,VAR
Rap	0.2762	**0.0642**	0.2095	A,5	0.0857	A,10
R'n'B	0.2571	**0.1458**	0.2238	A,VAR	0.2095	A,5
AdultContemporary	0.3818	**0.2417**	0.4045	A,10	0.3318	A,10
AlbumRock	0.2909	0.2316	0.2591	A,10	**0.1227**	C,5
AlternativePopRock	0.4039	**0.2251**	0.3063	C,5	0.2673	A,VAR
ClubDance	0.2594	**0.1760**	0.2925	C,VAR	0.2170	A,10
HeavyMetal	0.3750	0.1213	0.3705	C,VAR	**0.1205**	A,5
ProgRock	0.4886	0.2309	0.3508	C,VAR	**0.2080**	C,VAR
SoftRock	0.4535	**0.1862**	0.3093	C,5	0.4048	A,VAR
Urban	0.2130	0.2061	0.2348	A,VAR	**0.1652**	A,VAR

classification models: statistics of chords, harmonic properties like key and consonance level, shares of several recognised instruments, rhythmic and beat properties, model-based ensemble predictions of emotions, vocal characteristics, digital effects, etc., as well as the structural complexity of many semantic features (a method to measure temporal progress of feature time series after [31]). Essentially, as the approximative features are restricted to instrument and pitch properties, we can not expect that they can be the best for individual genres, but it seems that they are indeed comparable to the best selected features from a large semantic feature set, despite of a far from perfect performance to exactly recognise instruments in artificial polyphonic tracks. This suggests that a further optimisation of the instrument recognition performance may as well improve also the classification performance for genre prediction.

With regard to the last column of Table IV, all combinations of feature processing and fitness value estimation methods contribute to the best identified feature sets. This means that the best parameters depend on the classification task. When we fix the processing method and compare the best feature sets identified with $\phi \in \{5, 10, \text{VAR}\}$, there is no statistical difference between the errors (with the Wilcoxon signed rank test). When we fix the fitness estimation method, the only significant difference is observed for $\phi = 10$, here the "AR" processing seems to be better than "complete" processing, with p = 0.0437 which is rather close to the boundary value of 0.05. Therefore, we can not recommend a particular setting for these parameters which would perform at best for different genre categories.

V. CONCLUSIONS AND OUTLOOK

In this work, we have proposed a novel method to approximate polyphonic audio recordings with the help of an evolutionary algorithm which combines individual instrument tone samples and measures distances between candidate approximations and onsets of unknown recordings.

The first evaluation of instrument recognition performance in artificial tracks showed that the proposed method outperforms a random classifier, but still requires further improvement. However, in the second experiment, the best approximative features (similarities to instruments and pitches), identified with the help of evolutionary multi-objective feature selection for the recognition of music genres and styles, were comparable to the best selected semantic features from a large and diverse set which contained much more musically meaningful information than instrument and pitch properties only: those features comprised also temporal, structural, melodic, vocal, dynamics, emotional, and further characteristics, partly created with ensembles of supervised classifiers. For exactly one half of 14 genres and styles, our new approximative features were

better, and there was no statistical difference between the best selected features from both sets.

There exist a lot of possibilities to further improve the proposed algorithm which were beyond the scope of this study. In future, for instance, we plan to optimise feature domains for a better classification quality of recognised instruments, to add more musically meaningful genetic operators like the change of sample loudness or application of further signal augmentations, to apply classification ensembles, and to adjust the fitness evaluation which should better correlate with the instrument recognition performance.

ACKNOWLEDGEMENTS

This work was funded by the DFG (German Research Foundation, project 336599081 "Evolutionary optimisation for interpretable music segmentation and music categorisation based on discretised semantic metafeatures")

REFERENCES

[1] J. C. Brown, O. Houix, and S. McAdams, "Feature dependence in the automatic identification of musical woodwind instruments," *Journal of the Acoustical Soceity of America*, vol. 109, no. 3, pp. 1064–1072, 2001.

[2] A. Eronen, "Musical instrument recognition using ICA-based transform of features and discriminatively trained HMMs," in *Proceedings of the 7th International Symposium on Signal Processing and Its Applications (ISSPA)*, 2003, pp. 133 – 136.

[3] T. H. Park, *Introduction to Digital Signal Processing: Computer Musically Speaking*. Singapore: World Scientific, 2010.

[4] A. Zlatintsi and P. Maragos, "Multiscale fractal analysis of musical instrument signals with application to recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, no. 4, pp. 737–748, 2013.

[5] I. Vatolkin, M. Preuß, G. Rudolph, M. Eichhoff, and C. Weihs, "Multi-objective evolutionary feature selection for instrument recognition in polyphonic audio mixtures," *Soft Computing*, vol. 16, no. 12, pp. 2027–2047, 2012.

[6] T. Heittola, A. Klapuri, and T. Virtanen, "Musical instrument recognition in polyphonic audio using source-filter model for sound separation," in *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, 2009, pp. 327–332.

[7] Y. Han, J. Kim, and K. Lee, "Deep convolutional neural networks for predominant instrument recognition in polyphonic music," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 25, no. 1, pp. 208–221, 2017.

[8] J. S. Gómez, J. Abeßer, and E. Cano, "Jazz solo instrument classification with convolutional neural networks, source separation, and transfer learning," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 577–584.

[9] J. A. Biles, "GenJam: A genetic algorithm for generating jazz solos," in *Proceedings of the International Computer Music Conference (ICMC)*. San Francisco, USA: International Computer Association, 1994, pp. 131–137.

[10] I. Fujinaga and J. Vantomme, "Genetic algorithms as a method for granular synthesis regulation," in *Proceedings of the 1994 International Computer Music Conference (ICMC)*, 1994.

[11] R. De Prisco, D. Malandrino, G. Zaccagnino, and R. Zaccagnino, "An evolutionary composer for real-time background music," in *Proceedings of the 5th International Conference on Evolutionary and Biologically Inspired Music, Sound, Art and Design (EvoMUSART)*, C. Johnson, V. Ciesielski, J. Correia, and P. Machado, Eds. Cham: Springer International Publishing, 2016, pp. 135–151.

[12] E. R. Miranda and J. A. Biles, Eds., *Evolutionary Computer Music*. London, UK: Springer, 2007.

[13] , "Instrument sample database," 2020, accessed on 15.05.2020. [Online]. Available: https://ls11-www.cs.tu-dortmund.de/rudolph/mi#instrument_sample_database

[14] , "Best Service. Ethno World 5 Professional and Voices," 2010, accessed on 15.05.2020. [Online]. Available: https://www.youtube.com/watch?v=-9F3q8kAb00

[15] , "Native Instruments. Komplete 11 Ultimate," 2016, accessed on 15.05.2020. [Online]. Available: https://www.youtube.com/watch?v=WEfxP0-YZgQ

[16] T. Fujishima, "Realtime chord recognition of musical sound: a system using common lisp music," in *Proceedings of the International Computer Music Conference (ICMC)*, 1999, pp. 464–467.

[17] B. McFee, C. Raffel, D. Liang, D. P. E. nd Matt McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th Python in Science Conference*, 2015, pp. 18–25.

[18] A. Livshin and X. Rodet, "The significance of the non-harmonic "noise" versus the harmonic series for musical instrument recognition," in *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, 2006, pp. 95–100.

[19] D. Koelle, *The Complete Guide to JFugue: Programming Music in Java*, 2008, accessed on 15.05.2020. [Online]. Available: http://www.jfugue.org/4/jfbmrkklprpp/TheCompleteGuideToJFugue-v1.pdf

[20] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Upper Saddle River: Prentice Hall, 1993.

[21] D. McEnnis, C. McKay, I. Fujinaga, and P. Depalle, "jAudio: A feature extraction library," in *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, 2005, pp. 600–603.

[22] B. P. Bogert, M. J. R. Healy, and J. W. Tukey, "The quefrency analysis of time series for echoes: cepstrum, pseudo-autocovariance, cross-cepstrum and saphe-cracking," in *Proceedings of the Symposium on Time Series Analysis*, 1963, pp. 209–243.

[23] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 185–190, 1937.

[24] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler, "YALE: Rapid prototyping for complex data mining tasks," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, T. Eliassi-Rad, L. H. Ungar, M. Craven, and D. Gunopulos, Eds. ACM, 2006, pp. 935–940.

[25] M. Mauch and S. Dixon, "Approximate note transcription for the improved identification of difficult chords," in *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, J. S. Downie and R. C. Veltkamp, Eds., 2010, pp. 135–140.

[26] I. Vatolkin, G. Rudolph, and C. Weihs, "Interpretability of music classification as a criterion for evolutionary multi-objective feature selection," in *Proceedings of the 4th International Conference on Evolutionary and Biologically Inspired Music, Sound, Art and Design (EvoMUSART)*, 2015, pp. 236–248.

[27] , "OS120," 2020, accessed on 15.05.2020. [Online]. Available: https://ls11-www.cs.tu-dortmund.de/rudolph/mi#music_test_database

[28] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[29] , "Semantic feature list," 2020, accessed on 15.05.2020. [Online]. Available: https://ls11-www.cs.tu-dortmund.de/rudolph/mi/hl566

[30] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.

[31] M. Mauch and M. Levy, "Structural change on multiple time scales as a correlate of musical complexity," in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, A. Klapuri and C. Leider, Eds. University of Miami, 2011, pp. 489–494.