Mining Audio Data for Multiple Instrument Recognition in Classical Music

Elżbieta Kubera[1] and Alicja A. Wieczorkowska[2]

[1] University of Life Sciences in Lublin, Akademicka 13, 20-950 Lublin, Poland
elzbieta.kubera@up.lublin.pl
[2] Polish-Japanese Institute of Information Technology,
Koszykowa 86, 02-008 Warsaw, Poland
alicja@poljap.edu.pl

Abstract. This paper addresses the problem of identification of multiple musical instruments in polyphonic recordings of classical music. A set of binary random forests was used as a classifier, and each random forest was trained to recognize the target class of sounds. Training data were prepared in two versions, one based on single sounds and their mixes, and the other based on sound frames taken from classical music recordings. The experiments on identification of multiple instrument sounds in recordings are presented, and their results are discussed in this paper.

Key words: Music Information Retrieval, Sound Recognition, Random Forests

1 Introduction

Music information retrieval (MIR) became a topic of broad interest for researchers several years ago, see e.g. [14], [17], and one of the most challenging tasks within this area is to automatically extract meta-information from audio waveform [10]. Audio data stored as sound amplitude values changing over time represent very complex data, where multiple sounds of a number of instruments are represented by a single value (i.e. amplitude value of a complex sound) in each time instant in the case of monophonic recordings, or by a single value in each recording channel. Extraction of information about timbre of particular sounds is difficult, but it has been addressed in audio research last years. Identification of music titles through query-by-example, including excerpts replayed on mobile devices, has been quite successfully addressed [16], [19], as well as finding pieces of music through query-by-humming [11]. However, identification of instruments in audio excerpts is still a challenge, sometimes addressed through multi-pitch tracking [5], often supported with external provision of pitch data, or limited to identification of predominant sounds [2].

In this paper, we deal with identification of multiple sounds of multiple instruments in the recordings of classical music. No pitch tracking is required, and the classification is performed on data as is. There are no pre-assumptions regarding number of instruments in polyphony, and the recordings can contain

any instrument sounds, including instruments for which our classifiers are not trained. This is because we use a set of binary classifiers, where each classifier is trained to identify a target sound class. If none of the classifiers recognizes its target class, it means that an unknown instrument or instruments play in the analyzed audio sample. Classification is performed for mono or stereo input data, and a mix (average) of the channels is taken as input in the case of stereo recordings.

1.1 Audio Data Classification

Automatic identification of musical instruments has been performed so far by many researchers, and usually on a different set of instruments, number of classes, sound parametrization, number of sounds used, and classifiers applied. Identification of a single instrument in a single sound is the easiest case, and virtually all available classification tools have been applied for this purpose, including k-nearest neighbors, neural networks, support vector machines, rough set based classifiers, decision trees and random forests. Quality of the recognition depends heavily on the number of sounds and instruments/classes applied, and it can even reach 100% for a few classes, or be as low as 40% if there are 30 or more classes; for detailed review see [6].

Identification of instruments in polyphonic environment is much more challenging, and it has been addressed in various ways. Usually initial assumptions are made: on the number of instruments in the polyphony, on pitch data as input, on the instrument set in the analyzed recordings, or on identifying predominant sound, see e.g. [2], [5]. Since the final goal of such research is score extraction, such assumptions are understandable, and in some cases the research addresses sound separation into single sounds. However, these external data are often manually provided, not extracted from audio recordings. In our research, we would like to perform instrument recognition without any pre-assumptions, and also without initial data segmentation. We have already performed similar research, for jazz recordings [8], but it required tedious segmentation and labeling of small frames of the recordings in order to obtain ground-truth data. In order to facilitate research, we decided to perform annotation for 0.5-second excerpts; MIDI files and scores were used as guidance. Classification was performed using a set of random forests, since such a classifier proved quite successful in our previous research [8].

2 Random Forests

A random forest (RF) is a classifier based on a tree ensemble; such classifiers are gaining increasing popularity last years [15]. RF is constructed using procedure minimizing bias and correlations between individual trees. Each tree is built using a different N-element bootstrap sample of the N-element training set. The elements of the sample are drawn with replacement from the original set. Therefore, roughly 1/3 of the training data are not used in the bootstrap

sample for any given tree. Assuming that objects are described by a vector of K attributes (features), k attributes out of all K attributes are randomly selected ($k \ll K$, often $k = \sqrt{K}$) at each stage of tree building, i.e. for each node of any particular tree in RF. The best split on these k attributes is used to split the data in the node.

The best split of the data in the node is determined as minimizing the Gini impurity criterion, which is the measure of how often an element would be incorrectly labeled if labeled randomly, according to the distribution of labels in the subset. Each tree is grown to the largest extent possible, without pruning. By repeating this randomized procedure M times a collection of M trees is obtained, constituting a random forest. Classification of each object is made by simple voting of all trees [1].

The classifier used in our research consists of a set of binary random forests. Each RF is trained to identify the target sound class, representing an instrument (or silence), whether this particular timbre is present in the sound frame under investigation, or not. If the percentage of votes of the trees in the RS is 50% or more, then the answer of the classifier is considered to be positive, otherwise it is considered to be negative.

3 Audio Data

Our experiments focused on musical instrument sounds of definite pitch, but information about pitch was not used not retrieved in our research. Sounds of definite pitch are produced by chordophones (stringed instruments) and aerophones (wind instruments). In our experiments, we chose wind and stringed instruments, played in various ways, i.e. with various articulation, including bowing vibrato and pizzicato. Percussive instruments, i.e. idiophones and membranophones (basically drums) were excluded from the described research. Additionally, a class representing silence was added to the set of classes representing instruments. Therefore, if none of the classifiers gives positive answer, then we can conclude that an unknown instrument or instruments are playing in the investigated sound frame.

The following sound classes were investigated in the reported research:

- flute (*fl*),
- oboe (*ob*),
- bassoon (*bn*),
- clarinet (*cl*),
- French horn (*fh*),
- violin (*vn*),
- viola (*va*),
- cello (*cl*),
- double bass (*db*),
- piano (*pn*), and
- silence (*sc*).

All sounds were recording at 44.1 kHz sampling rate with 16-bit resolution, or converted to this format.

3.1 Training Data

Training of the classifiers was performed in two versions. The first training ($T1$) was based on single sounds of musical instruments, taken from RWC [4] sets of single sounds of musical instruments, and also mixes of up to three instrument sounds were added to this training set. Single sounds of musical instruments from MUMS [13] and IOWA [18] repositories were used for mixing. Ten thousands of 40-ms long audio frames represented positive examples for each RF (i.e. frames where the target instrument is playing), with 5,000 representing single sounds and 5,000 representing mixes, and ten thousands of 40-ms long audio frames represented negative examples (i.e. frames where the target instrument is not playing). Mixes constitute a single chord or unison, and a set of instruments is always typical for classical music.

The second training ($T2$) was based on sound taken from recordings, with no initial segmentation to separate single sounds. The recordings were taken from RWC Classical Music Database [3], recordings of Mozart concerto for Flute in G-major (KV 313), for Oboe in C-major (KV 314), for Bassoon in B-flat Major (KV 191), and the following .mp3 files (converted to .au format): viola Suite No. 1 in G-major BWV 1007, J.S. Bach, for cello solo transcribed for viola (Prelude and Allemande, [9]), Suite no 1 in G-major BWV 1007, J.S. Bach, for cello transcribed for double bass, Clarinet Concerto in A-major KV622, W.A. Mozart - Allegro and Adagio, Horn Concerto in E-flat Major KV 495, W.A. Mozart - Allegro moderato. Solo and polyphonic segments were taken as training data. This training set represents more realistic sounds, which can be encountered in all classical music recordings, including their compressed file version.

3.2 Testing Data

Testing was performed on RWC Classical Music Database recordings [3], and for presentation purposes the first minute of each investigated piece was used. The following pieces were used:

- No. 1, F.J. Haydn, Symphony no.94 in G major, Hob.I-94 'The Surprise'. 1st mvmt., with the following instruments playing in the first minute of the recording: flute, oboe, bassoon, French horn, violin, viola, cello, double bass;
- No. 2, W.A. Mozart, Symphony no.40 in G minor, K.550. 1st mvmt. with the following instruments playing in the first minute of the recording: flute, oboe, bassoon, French horn, violin, viola, cello, double bass;
- No. 16, W.A. Mozart, Clarinet Quintet in A major, K.581. 1st mvmt., with the following instruments playing in the first minute of the recording: clarinet, violin, viola, cello;
- No. 18, J. Brahms, Horn Trio in Eb major, op.40. 2nd mvmt., with the following instruments playing in the first minute of the recording: piano, French horn, violin;
- No. 44, N. Rimski-Korsakov, The Flight of the Bumble Bee, with flute and piano.

These pieces represent various polyphony and pose diverse difficulties for the classifier, including short sounds, and multiple instruments playing at the same time. No training data were used in our tests.

4 Feature Set

Audio data are usually parameterized before classification is applied, since raw data representing amplitude changes vs. time undergo dramatic changes in a fraction of second, and the amount of the data is overwhelming. The identification of musical instruments in audio data depends on the sound parametrization applied, and there is no one feature set used worldwide; each research group utilizes a different feature set. Still, some features are commonly used, and our feature set is also based on these features. We decided to utilize a feature set which proved successful in our previous, similar research [8]. Our parametrization is performed for 40-ms frames of audio data. No data segmentation or pitch extraction are needed, thus multi-pitch extraction is avoided, and no labeling particular sounds in polyphonic recording with the appropriate pitches is needed. The feature vector consists of basic features, describing properties of an audio frame of 40 ms, and additionally difference features, calculated as the difference between the feature calculated for a 30 ms sub-frame starting from the beginning of the frame, and a 30 ms sub-frame starting with 10 ms offset. Fourier transform was used to calculate spectral features, with Hamming window. Most of the features we applied represent MPEG-7 low-level audio descriptors, often used in audio research [7]. Identification of instruments is performed frame by frame, for consequent frames, with 10 ms hop size. Final classification result is calculated as an average of classifier output over 0.5-second segment of the recording, in order to avoid tedious labeling of ground-truth data over shorter frames.

The feature vector we applied consists of the following 91 parameters [8]:

- *Audio Spectrum Flatness*, $flat_1, \ldots, flat_{25}$ — a multidimensional parameter describing the flatness property of the power spectrum within a frequency bin for selected bins; 25 out of 32 frequency bands were used;
- *Audio Spectrum Centroid* — the power weighted average of the frequency bins in the power spectrum; coefficients are scaled to an octave scale anchored at 1 kHz [7];
- *Audio Spectrum Spread* — RMS (root mean square) value of the deviation of the log frequency power spectrum wrt. *Audio Spectrum Centroid* [7];
- *Energy* — energy (in log scale) of the spectrum of the parametrized sound;
- *MFCC* — a vector of 13 mel frequency cepstral coefficients. The cepstrum was calculated as the logarithm of the magnitude of the spectral coefficients, and then transformed to the mel scale, to better reflect properties of the human perception of frequency. 24 mel filters were applied, and the obtained results were transformed to 12 coefficients. The 13^{th} coefficient is the 0-order coefficient of MFCC, corresponding to the logarithm of the energy [12];

– *Zero Crossing Rate*; a zero-crossing is a point where the sign of the time-domain representation of the sound wave changes;
– *Roll Off* — the frequency below which an experimentally chosen percentage equal to 85% of the accumulated magnitudes of the spectrum is concentrated; parameter originating from speech recognition, where it is applied to distinguish between voiced and unvoiced speech;
– *NonMPEG7 - Audio Spectrum Centroid* — a linear scale version of *Audio Spectrum Centroid*;
– *NonMPEG7 - Audio Spectrum Spread* — a linear scale version of *Audio Spectrum Spread*;
– changes (measured as differences) of the above features for a 30 ms sub-frame of the given 40 ms frame (starting from the beginning of this frame) and the next 30 ms sub-frame (starting with 10 ms shift), calculated for all the features shown above;
– *Flux* — the sum of squared differences between the magnitudes of the DFT points calculated for the starting and ending 30 ms sub-frames within the main 40 ms frame; this feature by definition describes changes of magnitude spectrum, thus it is not calculated in a static version.

Audio data were in mono or stereo format; mixes of the left and right channel (i.e. the average value of samples in both channels) were taken if the audio signal was in stereo format.

5 Experiments and Results

The experiments aimed at investigating how many instruments can be identified correctly in real polyphonic recordings, and whether adding real recordings representing solos and polyphonic recordings (rather than isolated single sounds and their mixes) can improve the performance of the classifier. The main problem with such classification is the recall, which is usually quite low, i.e. instruments in recordings are missed by classifiers. Another problem is how to assess the results, since many instruments can be playing in the same segment. Since we deal with binary classifiers, possible errors include false negatives (missed target instrument) and false positives (false indication of the target instrument, not playing in a given segment). The details of classification results for both training versions, $T1$ and $T2$ are shown in Table 1 and Table 2. Precision, recall, f-measure and accuracy were calculated as follows, on the basis of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN):

– precision pr was calculated as $pr = TP/(TP + FP)$,
– recall rec was calculated as $rec = TP/(TP + FN)$,
– f-measure f_{meas} was calculated as $f_{meas} = 2 \cdot pr \cdot rec/(pr + rec)$,
– accuracy acc was calculated as $acc = (TP + TN)/(TP + TN + FP + FN)$.

If the denominator in the formula for calculating precision is equal to zero, then the classifier made no error, and the precision is equal to one. Also, if the

denominator in the formula for calculating recall is equal to zero, then the recall is equal to one.

As we can see, the classifier built for the training on single sounds and mixes ($T1$) gives few indications of the target classes; therefore, the precision is often equal to one (100%). The recall is quite low, but we are glad to observe that using real unsegmented recordings for training ($T2$) improves the recall significantly.

Table 1. Results of the recognition of musical instruments in RWC Classical Music Database, for training on single sounds and their mixes ($T1$)

Result	*bn*	*ob*	*cl*	*fl*	*fh*	*pn*	*cl*	*va*	*vn*	*db*	*sc*	Average
TP	2	1	6	21	0	23	18	14	0	22	17	
FP	52	0	0	0	5	3	5	11	0	13	40	
FN	114	98	41	153	168	207	192	227	394	115	2	
TN	432	501	553	426	427	367	385	348	206	450	541	
precision	4%	100%	100%	100%	0%	88%	78%	56%	100%	63%	30%	65%
recall	2%	1%	13%	12%	0%	10%	9%	6%	0%	16%	89%	14%
f-measure	2%	2%	23%	22%	0%	18%	15%	11%	0%	26%	45%	15%
accuracy	72%	84%	93%	75%	71%	65%	67%	60%	34%	79%	93%	72%

Table 2. Results of the recognition of musical instruments in RWC Classical Music Database, for training on sounds from real recordings, representing solos and polyphonic segments ($T2$)

Result	*bn*	*ob*	*cl*	*fl*	*fh*	*pn*	*cl*	*va*	*vn*	*db*	*sc*	Average
TP	54	7	4	150	23	50	142	99	327	65	17	
FP	90	9	218	87	85	25	58	85	3	45	32	
FN	62	92	43	24	145	180	68	142	67	72	2	
TN	394	492	335	339	347	345	332	274	203	418	549	
precision	38%	44%	2%	63%	21%	67%	71%	54%	99%	59%	35%	50%
recall	47%	7%	9%	86%	14%	22%	68%	41%	83%	47%	89%	47%
f-measure	42%	12%	3%	73%	17%	33%	69%	47%	90%	53%	50%	44%
accuracy	75%	83%	57%	82%	62%	66%	79%	62%	88%	81%	94%	75%

Illustration of the results for the first piece of music (No. 1) is shown in Figure 1 for the training $T1$, and for comparison the details of recognition results for the same piece for the training $T2$ is shown in Figure 2. As we can see, if the classifier trained on $T1$ (single sounds and mixes) shows positive outcome, the indication is just above the 0.5 threshold. The outcomes for the classifier trained on $T2$ (real solo and polyphonic recordings) are much higher, although we can see errors, especially for string instruments. However, this piece is difficult for recognition as an orchestral piece of high polyphony, so errors in this case were rather unavoidable.

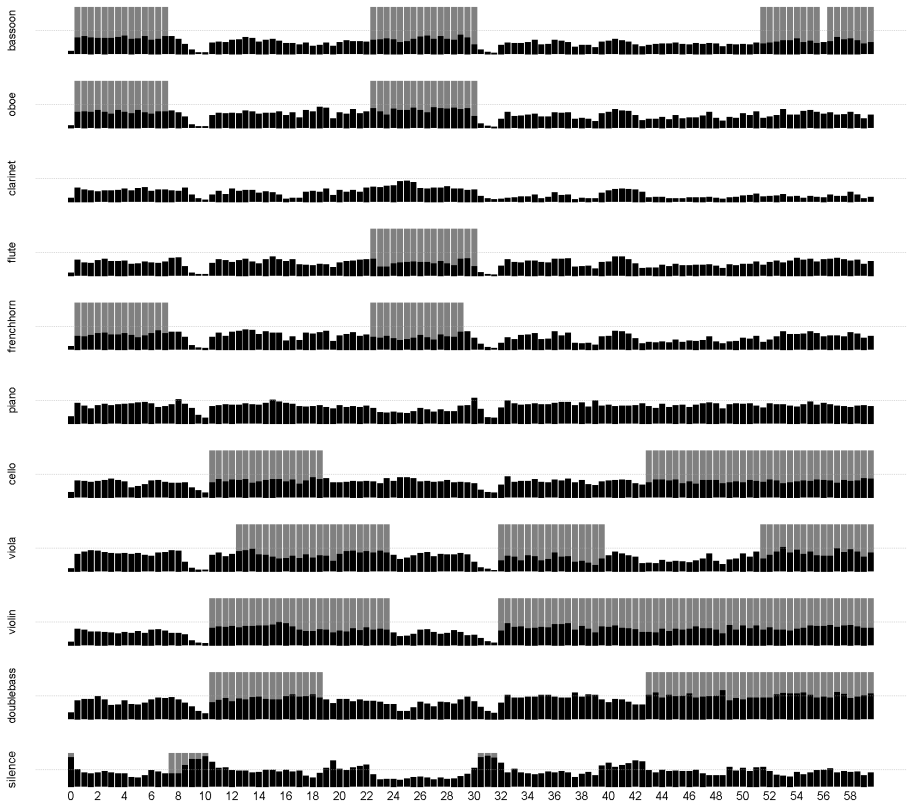

Fig. 1. Outcome of each random forest for the RWC Classical Music No. 1, for each 0.5-second segment of the first minute of the recording, for the training $T1$. If the result for a forest (trained to recognize a target instrument, or silence) is 0.5 or more, then this classifier indicates that the target instrument is playing in this segment. Ground-truth data are marked in grey.

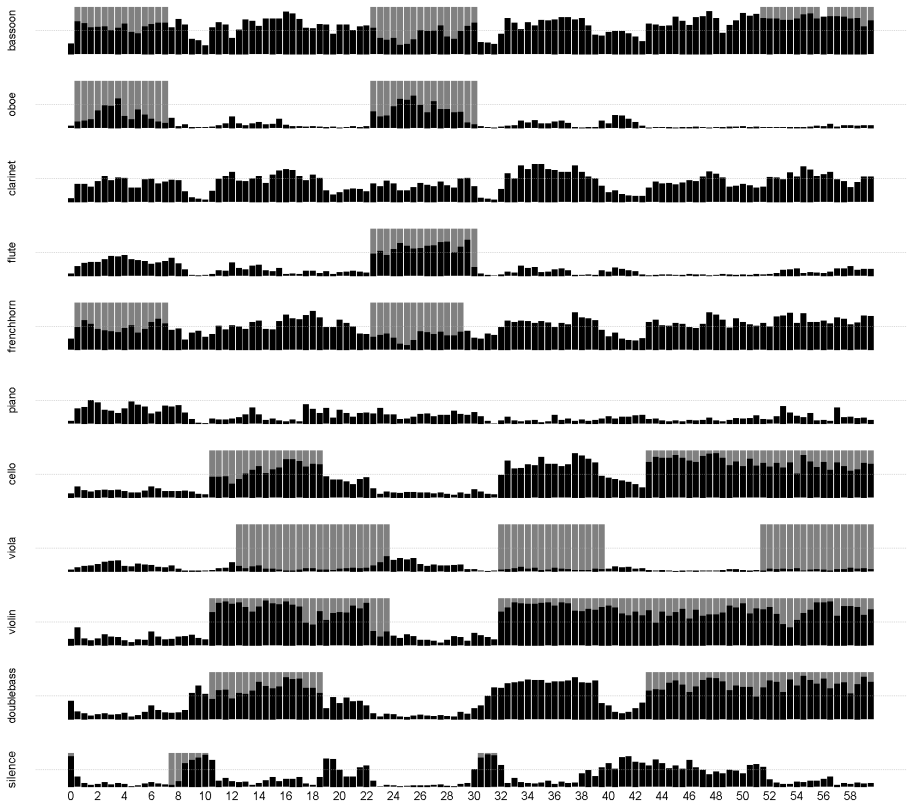

Fig. 2. Outcome of each random forest for the RWC Classical Music No. 1, for each 0.5-second segment of the first minute of the recording, for the training $T2$. If the result for a forest (trained to recognize a target instrument, or silence) is 0.5 or more, then this classifier indicates that the target instrument is playing in this segment. Ground-truth data are marked in grey.

6 Summary and Conclusions

The research presented in this paper aimed at the difficult task of recognizing multiple instruments in polyphonic recordings of classical music. Ten instrumental classes were investigated, and also silence was added as a separate class. The training was performed for single instrumental sounds and their mixes, and the second classifier was trained for excerpts taken from recordings, without segmentation. A set of binary random forests was used as a classifier, where each forest was trained to recognize whether the target instrument (or silence) is recorded in the analyzed excerpt. Forty-millisecond segments were analyzed, but the results were presented for 0.5-second segment, in order to avoid tedious labeling of ground-truth data. The results show that training performed on unsegmented real recordings improves the recall dramatically. Therefore we conclude that the training data should be adjusted to the target in hand, thus allowing the classifier to learn the sounds in typical recording set-up.

Acknowledgments. This project was partially supported by the Research Center of PJIIT, supported by the Polish Ministry of Science and Higher Education.

References

1. Breiman, L.: Random Forests. Machine Learning 45, 5–32 (2001)
2. Fuhrmann, F.: Automatic musical instrument recognition from polyphonic music audio signals. PhD Thesis, Universitat Pompeu Fabra (2012)
3. Goto, M., Hashiguchi, H., Nishimura, T., Oka, R.: RWC Music Database: Popular, Classical, and Jazz Music Databases. In: Proceedings of the 3rd International Conference on Music Information Retrieval, 287–288 (2002)
4. Goto, M., Hashiguchi, H., Nishimura, T., Oka, R.: RWC Music Database: Music Genre Database and Musical Instrument Sound Database. In: Proceedings of the 4th International Conference on Music Information Retrieval ISMIR, 229–230 (2003)
5. Heittola, T., Klapuri, A., Virtanen, A.: Musical Instrument Recognition in Polyphonic Audio Using Source-Filter Model for Sound Separation. In: Proc. 10th Int. Society for Music Information Retrieval Conf. (ISMIR 2009) (2009)
6. Herrera-Boyer, P., Klapuri, A., Davy, M.: Automatic Classification of Pitched Musical Instrument Sounds. In: Klapuri, A., Davy, M. (eds.) Signal Processing Methods for Music Transcription. Springer Science+Business Media LLC (2006)
7. ISO: MPEG-7 Overview, `http://www.chiariglione.org/mpeg/`
8. Kubera, E., Kursa, M.B., Rudnicki, W.R., Rudnicki, R., Wieczorkowska, A.A.: All That Jazz in the Random Forest. In: Kryszkiewicz, M., Rybiński, H., Skowron, A., Raś, Z.W. (eds.): ISMIS 2011. LNAI, vol. 6804, pp. 543–553. Springer, Heidelberg (2011)
9. Kuperman, M.: Suite N 1 in G-Dur BWV 1007, `http://www.viola-bach.info/`
10. Martin, K.D.: Toward automatic sound source recognition: Identifying musical instruments. Presented at the 1998 NATO Advanced Study Institute on Computational Hearing, Il Ciocco, Italy (1998)
11. MIDOMI: Search for Music Using Your Voice by Singing or Humming, `http://www.midomi.com/`

12. Niewiadomy, D., Pelikant, A.: Implementation of MFCC vector generation in classification context. J. Applied Computer Science, Vol. 16, No. 2, pp. 55–65 (2008)
13. Opolko, F., Wapnick, J.: MUMS — McGill University Master Samples. CD's (1987)
14. Ras, Z.W., Wieczorkowska, A.A. (eds.): Advances in Music Information Retrieval. Series: Studies in Computational Intelligence, Vol. 274, Springer (2010)
15. Richards, G., Wang, W.: What influences the accuracy of decision tree ensembles? J. Intell. Inf. Syst. 39, 627-650 (2012)
16. Shazam Entertainment Ltd, http://www.shazam.com/
17. Shen, J., Shepherd, J., Cui, B., Liu, L. (eds.): Intelligent Music Information Systems: Tools and Methodologies. Information Science Reference, Hershey (2008)
18. The University of IOWA Electronic Music Studios: Musical Instrument Samples, http://theremin.music.uiowa.edu/MIS.html
19. TrackID — Sony Smartphones, http://www.sonymobile.com/global-en/support/faq/xperia-x8/internet-connections-applications/trackid-ps104/