GOOD-SOUNDS.ORG: A FRAMEWORK TO EXPLORE GOODNESS IN INSTRUMENTAL SOUNDS

Giuseppe Bandiera[1] **Oriol Romani Picas**[1] **Hiroshi Tokuda**[2]
Wataru Hariya[2] **Koji Oishi**[2] **Xavier Serra**[1]
[1] Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
[2] Technology Development Dept., KORG Inc., Tokyo, Japan
giuseppe.bandiera@upf.edu, oriol.romani@upf.edu

ABSTRACT

We introduce good-sounds.org, a community driven framework based on freesound.org to explore the concept of goodness in instrumental sounds. Goodness is considered here as the common agreed basic sound quality of an instrument without taking into consideration musical expressiveness. Musicians upload their sounds and vote on existing sounds, and from the collected data the system is able to develop sound goodness measures of relevance for music education applications. The core of the system is a database of sounds, together with audio features extracted from them using MTG's Essentia library and user annotations related to the goodness of the sounds. The web front-end provides useful data visualizations of the sound attributes and tools to facilitate user interaction. To evaluate the framework, we carried out an experiment to rate sound goodness of single notes of nine orchestral instruments. In it, users rated the sounds using an AB vote over a set of sound attributes defined to be of relevance in the characterization of single notes of instrumental sounds. With the obtained votes we built a ranking of the sounds for each attribute and developed a model that rates the goodness for each of the selected sound attributes. Using this approach, we have succeeded in obtaining results comparable to a model that was built from expert generated evaluations.

1. INTRODUCTION

Measuring sound goodness, or quality, in instrumental sounds is difficult due to its intrinsic subjectivity. Nevertheless, it has been shown that there is some consistency among people while discriminating good or bad music performances [1]. Furthermore, recent studies have demonstrated a correlation between the perceived music quality and the musical performance technique [2]. Bearing this in mind, in a previous work [3] we proposed a method to automatically rate goodness by defining a set of sound attributes and by using a set of good/bad labels given by expert musicians. The definition of goodness was treated as a classification problem and an outcome of that work was a mobile application (Cortosia ®) that gives goodness scores in real-time for single notes on a scale from 0 to 100. This score was computed considering the distribution of the features values in the classification step. While developing that system we realized that we could improve the scores, specially their correlation with the perceptual sound goodness, if we could use more training data and include a range of goodness levels given by users rather than the binary good/bad labels that we used. However, the task of labeling sounds this way would have been very time consuming and we would also need more sounds, covering the whole range of sound goodness. To address these issues we are now crowdsourcing the problem. We have developed a website, good-sounds.org, on which users can upload sound content and can tag and rate sounds in various ways.

2. GOOD-SOUNDS.ORG

Good-sounds.org [1] is an online platform to explore the concept of goodness in instrumental sounds with the help of a community of users. It provides social community features in the web front-end and a framework for sound analysis and modeling in the background. It also includes an API to access the collected data.

2.1 Description

The website has been designed from a user perspective, meant to be modern and to provide a seamless experience. It makes use of state of the art design concepts and community oriented web technologies. The web front-end includes three main sections: (1) a page to list and visualize the uploaded sounds as shown in Figure 1, (2) a page to upload and describe sounds as shown in Figure 2 and (3) a section to gather user ratings and annotations. The visualization page shows a list of all the sounds and it includes filter options to narrow down the results, being able to show things like specific instruments or sounds uploaded a certain date. The upload page allows users to add sounds into the site and also provides a recording tool

[1] *https://good-sounds.org*

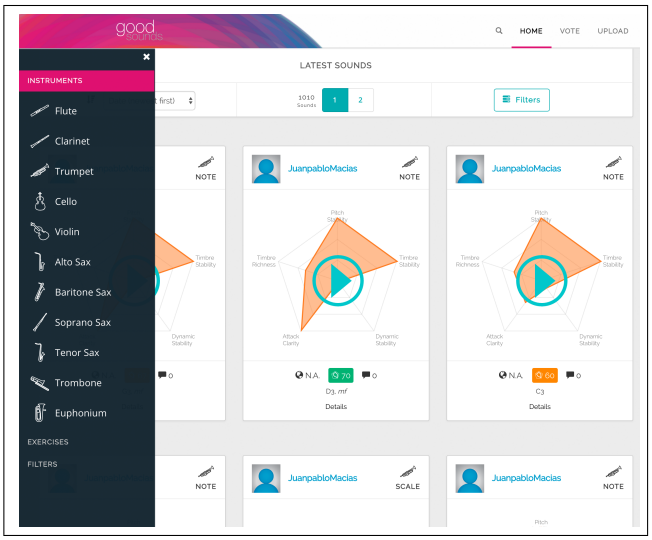

Figure 1. Good-sound.org sound list page.

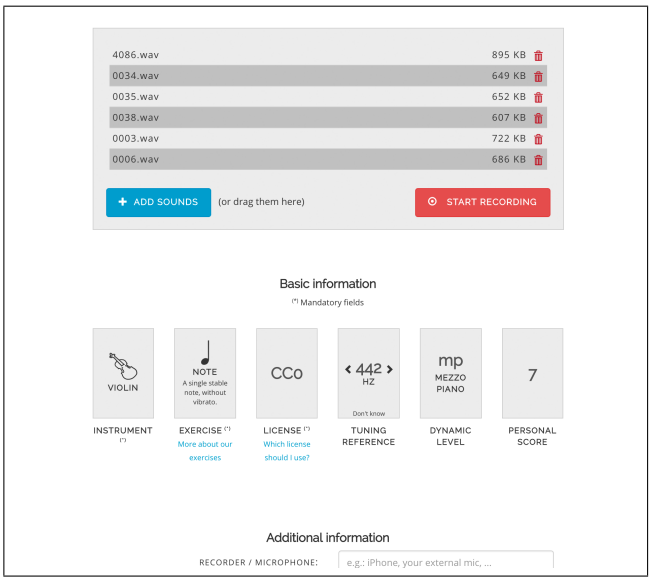

Figure 2. Good-sound.org sound upload page.

built using Web Audio API [2] . The annotation section has been designed for the specific experiment explained in Section 3. The website backend is based on the experience we have obtained in all these years developing and maintaining freesound [2] . It is written in Python using the Django web application framework. The metadata is stored in a PostgreSQL database while the sound files plus other analysis files are stored locally in the server. An API accepts requests from authorized clients to upload sounds (currently through the mobile app Cortosia) and retrieve statistics from the users community. At this time, the website supports 11 instruments, it includes 8470 unique sounds and there are 363 active users.

2.2 Content

The main data stored in good-sounds.org consists of sounds and the metadata accompanying them. When uploading sounds, the users can choose between three different types of Creative Commons licenses for their content: Universal, Attribution or Attribution Non-Commercial. As soon as a sound is uploaded, it is analyzed using the freesound extractor [4], thus obtaining a number of low-level audio features, and the system generates an mp3 version of the file together with images of the waveform and spectrogram. The audio, image and audio feature files are stored in the good-sounds server and the metadata is stored in the PostgreSQL database.

2.2.1 Segmentation

One of the critical audio processing steps performed in good-sounds.org is the segmentation of the uploaded sound files to find appropriate note boundaries. Given that the audios come from different and not well controlled sources, they might include all kinds of issues (ex. silence at beginning and end or background noise) that can difficult the subsequent feature extraction steps. Considering that the sounds we are working with are all monophonic pitched instrument sounds, we can base the segmentation mainly on pitch. Our approach extracts pitch using Essentia's [5] implementations of the YinFFT algorithm [8] and the Yin time based algorithm [6]. Then the sound is segmented into notes using pitch contours [7] and signal RMS with Essentias PitchContourSegmentation algorithm. The segmentation data is also stored in the database. This allows us to build client-side data visualizations that effectively reflect the quality of the segmentation algorithm and the user can modify the parameters for this algorithm and re-run it on the fly from the website. The results of this iteration is immediately shown on the same page, for an easy comparison of the results, as it is shown in Figure 3.

Figure 3. Good-sound.org segmentation visualisation page.

2.2.2 Descriptors

The feature extraction module is based on the freesound extractor module of Essentia. It computes a subset of its spectral, tonal and temporal descriptors. With it, the audios are first resampled to 44.1kHz sampling rate. The descriptors are then extracted across all the frames using a 2048 window size and 512 hop window size. We then compute statistical measures (mean, median and standard deviation)

[2] *http://www.w3.org/TR/webaudio/*

of the descriptors which are the values stored as JSON files in the server. The list of the descriptors extracted is shown in Table 1.

3. EXPERIMENT

As a test case to evaluate the usefulness of the good-sounds.org framework we setup an experiment to rate the goodness of single notes. The goal of the experiment was to build models from both the uploaded sounds and the community annotations, with which we can then automatically rate the sound goodness. We compared the results of the obtained models with the ones we got in our previous work using expert annotations.

3.1 Dataset

The data used in this experiment comes from several sources. First, we uploaded all the sounds from our previous work to the website, together with the expert annotations. Since the website has been public for a few months, we also had sounds uploaded by users, both directly and through the mobile app (using the API). Then, user annotations on the sounds according to a goodness scale where collected using a voting task. These annotations use a set of sound attributes that affect sound goodness. These attributes were defined in our previous article [3] by consulting with a group of music experts:

- *dynamic stability*: the stability of the loudness.
- *pitch stability*: the stability of the pitch.
- *timbre stability*: the stability of the timbre.
- *timbre richness*: the quality of the timbre.
- *attack clarity*: the quality of the attack.

The sounds from the recording sessions are also uploaded to freesound and thus are openly accessible. We also provide a tool that allow the good-sounds.org users to link their accounts to their freesound ones and upload the sounds there.

3.1.1 Sounds

For this experiment we only used single note sounds. At the time of the experiment there were sounds for 5467 single notes of 9 instruments. We show the list of sounds per instrument in Table 2. The sounds we recorded ourselves from the recording sessions are uncompressed wave files, while the ones uploaded by users to the website are in different audio formats.

3.1.2 Annotations

We distinguish two kinds of annotations: (1) recording annotations and (2) community annotations. The recording annotations are the ones coming from the recording sessions that we did and consists of one tag per sound. This tag says if the sound is a good or a bad example of each sound attribute (e.g. bad-timbre-stability, good-attack-clarity). Those are the annotations used later on for a first evaluation of the models and are only available for the sounds we recorded ourselves. The community annotations are the ones generated from the user votes and used in this work to explore goodness. In order to be able to rate a sound in a goodness scale we need annotations on a wide range of different goodness levels. We originally thought of asking the community to rate sounds in a scale of goodness but we discarded this option because of the following:

instrument	number of sounds
cello	935
violin	802
clarinet	1360
flute	1434
alto sax	352
baritone sax	292
tenor sax	292
soprano sax	343
trumpet	738

Table 2. Number of sounds in the experiment's dataset.

- the task can be excessively demanding.
- without a reference sound the criteria of different users can differ extremely.
- with a reference sound we influence the users criteria, thus annotations can be less generalisable.

Instead, we designed a user task that gave as outcome a ranked list of the sounds based on the goodness for each sound attribute. An A/B multi vote task was used for this purpose. Two sounds are presented and the user is asked to decide which sound is better according to one or more of the sound attributes. One vote is stored for each selected attribute. A list of the votes per instruments (considering all sound attributes) is shown in Table 3. In order to prevent random votes in the task we run checks periodically. This checks consists of two sounds; one being a bad example of a sound attribute regarding the expert annotations and the other being a good example. The task is presented to the users the first time they vote and also randomly after some votes. If the user does not vote for the expected sound in the reference task, his next votes are not used. The votes of this user are again taken into account if he succeeds to pass the reference task.

3.1.3 Rankings

In order to have learning data in a wide range of goodness we built rankings with the community votes for each sound attribute. The position of a sound in the ranking represents its goodness level. To build them we count the number of wins and the number of votes of each sound in the database. Then the sounds are sorted according to two parameters:

- total number of votes: number of participations in the voting task.

spectral	tonal	temporal
spectrum, barkbands, melbands, flatness, crest, rolloff, decrease, hfc, pitch salience, flatness db, skewness, kurtosis, spectral complexity, flatness sfx,	pitch yinfft, pitch yin, pitch confidence,	zerocrossingrate, loudness, centroid,

Table 1. Descriptors extracted by Essentia library present in good-sounds.org.

instrument	number of votes
cello	140
violin	90
clarinet	293
flute	305
alto sax	78
baritone sax	59
tenor sax	14
soprano sax	21
trumpet	230

Table 3. Number of votes in good-sounds for the dataset's experiment.

- ratio between wins and votes: the ratio between the number of wins and the total number of participations in the voting task.

Using these parameters for building the rankings we assure that the sounds in the top are the ones voted more times, as being better than others, and not sounds with few votes but high percentage of wins.

3.2 Learning

The goal of our learning process is to build a model for each instrument that is able to rate each sound attribute in a 0 to 100 score. To do so we want to find a set of features that highly correlate with the rankings extracted in the previous step. Our approach uses a regression model to predict the score. These predictions are then used as samples of the final score function. The final score is then computed as an interpolation of the samples.

3.2.1 Models

We want to find the combination of regression model and set of features that better describes the rankings. For such a purpose we tried different regression algorithms available in scikit-learn [9]. As one of the outputs of the project is a system that rates the goodness of sounds in real-time we want to restrict the number of features in order to maintain a low computational cost. For each one of the algorithms we build a model for each ranking using one, two or three features and we compute the average prediction score of the model across all the options. The prediction score R^2 or Coefficient of Determination is defined as follows:

$$R^2 = (1 - u/v) \tag{1}$$

where

$$u = \sum((y_{true} - y_{pred})^2) \tag{2}$$

Regression model	Avg. score	Score variance
SVR 3 features	-1.208	5.4436
SVR 2 features	-1.2411	5.3895
SVR 1 feature	-1.4254	5.6554

Table 5. Performance of SVR model with different number of features.

and

$$v = \sum((y_{true} - \prod_{i=1}^{n} y_{true})^2) \tag{3}$$

Where y_{true} is the set of ground truth annotations and y_{pred} the set of predictions, having both the same length. The best possible score R^2 is 1.0 and it can be negative. The variance of the prediction score across all the rankings and set of features is also computed. The number of features that give the best score for each ranking is taken into account to compute an average number of features for each regression model. A comparison of the performance of the different models is shown in Table 4.

As we can see in the table, the SVR (Epsilon-Support Vector Regression) model has the best average score across all the rankings and using all possible combination of feature sets (up to 3 features). It also has the lowest score variance so we can expect the model to be robust across the different instruments and sound attributes. However the average number of features is almost two and the computation of two features at each frame of all the sounds in the database can be computationally expensive. For this reason we tested how good the model can be if we force it to use less than three features. We show the results of such a comparison in Table 5.

The results show that the differences between using one or three features are not too big so we decided to use SVR with a single feature in order to maintain a low computational cost for future applications of the system. We then tried all possible combinations of parameters (kernel, degree of polynomially, cost parameter..) to find the best model for each instrument and sound attribute.

3.2.2 Scores

From the model we predict the ranking position of a sound and we map this position into a 0 to 100 score of the sound attribute. The final goodness score is computed as the average score across the five attributes. We compute the sound attribute scores of all the sounds in the database to test the distribution of the scores according to the feature value. For example, a distribution of the score for the timbre stability of flute is shown in Figure 4.

Regression model	Avg. score	Score variance	Average of features
SVR	-1.208	5.4436	1.843
Ridge	-2.644	31.005	2.166
KRR	-1.79	10.798	1.906
Linear regression	-3.503	30.03	1.718
RANSAC	-3.202	17.532	1.478
Theilsen	-4.14	37.135	1.781

Table 4. Performance of the different regression models.

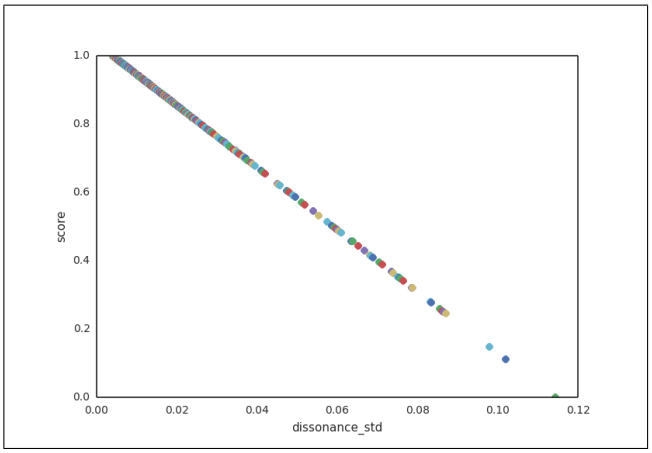

Figure 4. Distribution of scores of flute timbre stability.

The resulting distributions are not balanced. For this reason we push the scores of each sound attribute to fit a Gaussian distribution. This gives us balanced distributions and it also allows us to refine the scores by tweaking the parameters of the gaussian function. A result of this process is shown in Figure 5. The final score is computed interpolating the feature according to these tuned distributions.

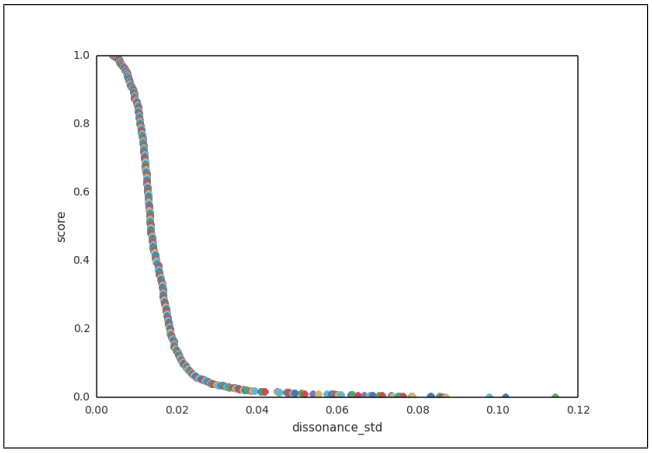

Figure 5. Distribution of scores of flute timbre stability after normalisation.

3.2.3 Models evaluation

In order to evaluate the models we want to check the correlation between the scores and the rankings as we expect the sounds ranked in the first positions to have the highest scores. We evaluate this correlation using Kendall Rank Correlation Coefficient [10], commonly referred as Kendalls tau coefficient. We use the implementation available in the scipy library, that is based on the tau-b version. Its computation, given two rankings x and y of the same size is defined by the following equation:

$$\tau = \frac{(P - Q)}{\sqrt{((P + Q + T) * (P + Q + U))}} \quad (4)$$

where P is the number of concordant pairs, Q the number of discordant pairs, T the number of ties only in x, and U the number of ties only in y. If a tie occurs for the same pair in both x and y, it is not added to either T or U. The values range from -1 to 1, where 1 indicates strong agreement and -1 strong disagreement. We compute τ between the score and the ranking position for all the sounds that are contained in the rankings. The results for each sound attribute and instrument are shown in Table 6.

4. CONCLUSIONS

In this article we presented a web based framework for exploring sound goodness in instrumental sounds using a community of users. The framework provides an easy way to collect sounds and annotations as well as tools to extract and store music descriptors. This allows us to explore the concept of sound goodness in a controlled and flexible environment. Furthermore, the website is useful to the community as a place in which to discuss the issues affecting sound goodness as well as a learning tool to improve their playing techniques. As a way to evaluate the framework we extended our previous work by using annotations from the community collected through a voting task. The models built using this approach provide an automatic rating of goodness for each attribute that tends to match the expert annotations collected in our previous work. The results should improve with more annotations from the community. As future work we want to design new tasks to collect user annotations and build new models according to them.

5. ACKNOWLEDGEMENTS

This research has been partially funded by KORG Inc. The authors would like to thank the entire good-sounds.org community who contributed to the website with sounds and annotations.

Sound attribute	Flute	Violin	Clarinet	Trumpet	Cello	Violin	Alto sax	Baritone sax	Soprano	**Average**
timbre stability	0.37	0.65	0.46	0.38	0.28	0.65	0.33	0.73	0.73	**0.51**
dynamic stability	0.41	0.33	0.24	0.44	0.64	0.33	0.33	0.31	0.31	**0.37**
pitch stability	0.42	0.46	0.22	0.38	0.58	0.46	1	1	0.81	**0.59**
timbre richness	0.04	0.35	0.32	0.11	0.21	0.35	1	0.56	1	**0.43**
attack clarity	0.33	0.59	0.38	0.3	0.18	0.59	0	0.34	0.35	**0.34**

Table 6. Kendall tau coefficient between the scores and the rankings of each sound attribute.

6. REFERENCES

[1] J. Geringer and C. Madsen. "Musicians ratings of good versus bad vocal and string performances," *Journal of Research in Music Education*, vol. 46, pages 522-534, 1998.

[2] Brian E. Russell. "An empirical study of a solo performance assessment model," *International Journal of Music Education*, vol. 33, pages 359-371, 2015.

[3] O. Roman Picas, H. Parra Rodriguez, D. Dabiri, H. Tokuda, W. Hariya, K. Oishi, and X. Serra. "A real-time system for measuring sound goodness in instrumental sounds," *Audio Engineering Society Convention 138.* Audio Engineering Society 2015.

[4] F. Font, G. Roma, and X. Serra. "Freesound technical demo," *Proceedings of the 21st ACM international conference on Multimedia*, 2013.

[5] D. Bogdanov, N. Wach, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. R. Zapata, and X. Serra. "Essentia: An audio analysis library for music information retrieval," *Proceedings of the International Society for Music Information Retrieval Conference,* pages 493-498, 2013.

[6] A. de Cheveign and H. Kawahara. "YIN, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America,* pages 111-1917, 2002.

[7] R. J. McNab, Ll. A. Smith, and I. H. Witten. "Signal processing for melody transcription," *Australasian Computer Science Communications 18*, pages 301-307, 1996.

[8] P. M. Brossier. "Automatic Annotation of Musical Audio for Interactive Applications," *Ph.D. Thesis, Queen Mary University of London, UK*, 2007.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas. "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research 12*, pages 2825-2830, 2011.

[10] Stepanov, Alexei. "On the Kendall Correlation Coefficient," arXiv preprint arXiv:1507.01427, 2015.