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 framework to explore the concept of goodness in instrumental sounds. Musicians can upload their sounds and vote on existing sounds, and from that 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 and user annotations related to the goodness of the sounds. The features are extracted using various algorithms from the Essentia software library and the web front-end provides useful data visualisations of the sound attributes and tools to facilitate user interaction. To evaluate the framework we carried out an experiment to rate sound goodness on single notes of nine different instruments. Users rated the sounds using an AB vote over a set of sound attributes defined for characterizing single notes of instrumental sounds. With the obtained votes we built a ranking of the sounds for each attribute and then from the ranking we 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 and 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 realised 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 crowsourcing the problem. We have developed a website, good-sounds.org, on which users can upload sound content and can tag sounds in various ways.

2. GOOD-SOUNDS.ORG

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

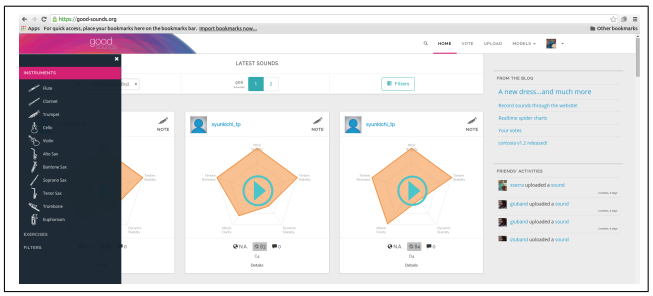

Figure 1. Good-sound.org sound list page.

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 designing concepts and community oriented web technologies. The front-end includes

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

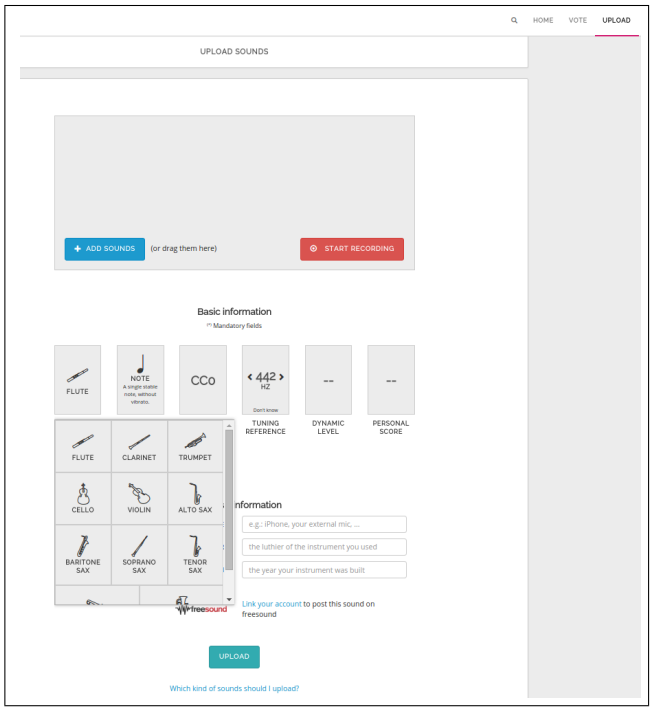

Figure 2. Good-sound.org upload page.

three main sections: (1) a visualisation page for the uploaded sounds that is shown in Figure 2, (2) a page to upload and describe sounds shown in Figure **??** and (3) and a section to gather user annotations. The visualisation page shows a list of all the sounds in the database that can be filtered by several options like instrument or uploading date. The upload page allows the users to add or drag sounds into the site and also provides a recording tool built using Web Audio API [2] . The annotation section has been designed for the specific experiment explained in section 3. The website backend is written in Python using the Django web application framework. The metadata is stored in a PostgreSQL database while the 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 core data stored in good-sounds.org consists of sounds and the metadata accompaning 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] *http://www.w3.org/TR/webaudio/*

2.2.1 *Segmentation*

One of the critical audio processing steps performed in good-sounds.org is the segmentation of the uploaded audio files to find appropiate note boundaries. As the audios come from different and not well controlled sources, they might include all kinds of issues (ex. silence at begining and end, background noise, ...) that can difficult the subsequent feature extraction step. Considering that the sounds we are working with are all monophonic pitched instrument sounds, we can base the segmentation on pitch. Our approach extracts pitch using Essentias [5] implementations of the YinFFT algorithm [6] and the Yin time based algorithm [7]. Then the sound is segmented into notes using pitch contours [8] and signal RMS with Essentias PitchContourSegmentation algorithm. All 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. Moreover, the user can modify the parameters for this algorithm and rerun 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.

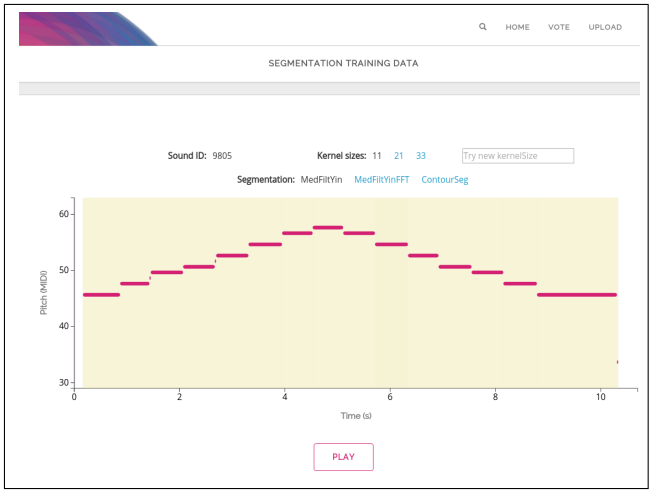

Figure 3. Good-sound.org segmentation visualisation page.

2.2.2 *Descriptors*

The feature extraction module used [4] computes spectral, tonal and temporal descriptors. The audios are resampled to 44.1kHz sampling rate and normalised using replay gain. 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) of the descriptors which are the values stored as JSON files in the server. A list of the descriptors we extract 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

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

to build models from the uploaded sounds and the community annotations with which we can then automatically rate the sound goodness. We then compared the results of these models with the ones we obtained in our previous work using the 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. The website has been public for a few months, thus we had the sounds uploaded by the users, both directly and through the mobile app (using the API). Finally, user annotations on the sounds according to a goodness scale where collected using a voting task. These annotations are taken using to a set of sound attributes that affect sound goodness. We defined these attributes in our previous work together 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.

3.1.1 Sounds

For this experiment we selected only the 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.

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.

3.1.2 Annotations

We distinguish two kind 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:

- 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. On 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 sound is expected in the reference task, his next votes are not considered. The votes of this user are again taken into account if he succeeds on 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

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.

the database. Then the sounds are sorted according to two parameters:

- total number of votes: number of participations in the voting task.
- 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 use all the different regression algorithms available in scikit-learn [10]. 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 is defined in the scikit-learn documentation as follows:

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

where

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

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.

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 differnet number of features.

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 want to know 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 use of more than one feature dos not give significally better results so we decided to use SVR with a single feature. We then tried all possible combinations of parameters (kernel, degree of polinomiality, cost parameter..) to find the best model for each instrument and sound attribute.

3.2.2 Scores

From the model we are able to predict the ranking position of a sound. We then 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.

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.

3.2.3 Models evaluation

In order to evaluate if the scores provided by the models fit the perceived goodness, we used the expert annotations of our previous work. These annotations only describe if the sound is a good or a bad example of one of the sound attributes defined previously. A model for each

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.

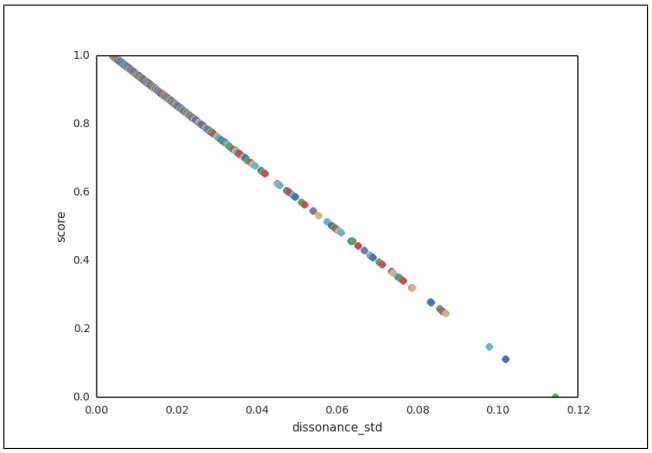

Figure 4. Distribution of scores of flute timbre stability.

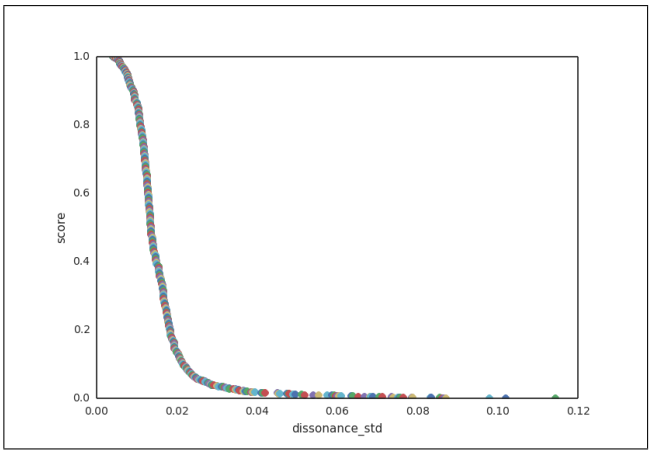

Figure 5. Distribution of scores of flute timbre stability after normalisation.

sound attribute of each instrument is trained using the rankings described previously. These rankings are built with all the sounds from which we have votes (around 20% of the sounds in the dataset). We use the models to compute the scores of all the sounds in the dataset and then we use the data visualisation tools implemented in the website to visualise the scores according to the expert annotations. An example of this using flute timbre stability is shown in Figure 4. The visualisation tools read from database so just by training models the results are instantly shown in the web page.

We expect the scores of good examples to be higher than 50 and the bad ones lower than 50. The number of sounds per sound attribute with scores above and below 50 accord-

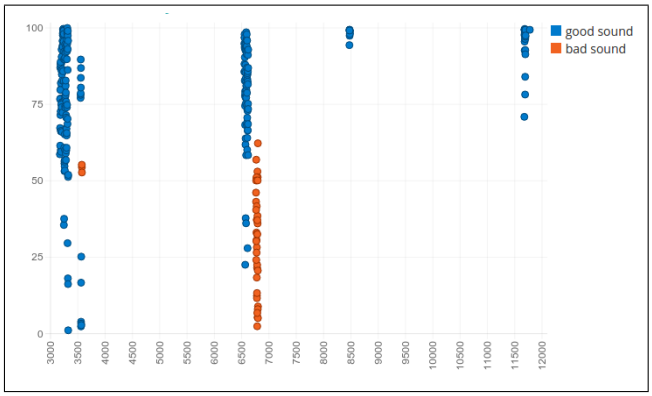

Figure 6. Visualisation of scores through the website tools.

ing to their tags are shown in Table 6.

As we can see, the models seem to perform better in rating good sound examples. We can consider as correct predictions the sounds with good sound tag and score ¿ 50 as well as the ones having bad attribute tag and score ¡ 50. Thus there are 8778 correct predictions out of 12427. As future work we want to also evaluate the models by using community annotations on scores.

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 the work presented in [1] 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.

Dynamics stability		
sound tag	score >= 50	score <50
good sound	1846	872
bad dynamics	438	881

Timbre stability		
sound tag	score >= 50	score <50
good sound	930	429
bad timbre	224	472

Pitch stability		
sound tag	score >= 50	score <50
good sound	1260	99
bad pitch	125	636

Timbre richness		
sound tag	score >= 50	score <50
good sound	1184	175
bad richness	380	268

Attack clarity		
sound tag	score >= 50	score <50
good sound	1005	354
bad attack	553	294

Table 6. Number of sounds for each sound attribute with scores higher or lower than 50.

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.

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 522-34, 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. Romani Picas et al. "A real-time system for measuring sound goodness in instrumental sounds," *AES 138th Convention Warsaw*, 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 et al. "Essentia: An audio analysis library for music information retrieval," *In Proceedings of the International Society for Music Information Retrieval Conference.*pages 493498, 2013.

[6] P. M. Brossier "Automatic Annotation of Musical Audio for Interactive Applications" *QMUL, London, UK,*, 2007.

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

[8] R. J. McNab et al. "Signal processing for melody transcription" *In Proceedings of the 19th Australasian Computer Science Conference*, 1996.

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

[10] Pedregosa et al. "Scikit-learn: Machine Learning in Python" *JMLR 12*, pages. 2825-2830, 2011.