JSI Sound – a machine-learning tool in Orange for simple biosound classification

Martin Gjoreski, Borut Budna, Anton Gradišek, Matjaž Gams
Department of Intelligent Systems, Jožef Stefan Institute
Jamova cesta 39, SI-1000 Ljubljana
anton.gradisek@ijs.si

Abstract

The recent advances in machine learning (ML) and sound processing have considerably expanded the possibilities provided by such techniques. The next step are open-access tools in existing ML toolkits for general sound ML tasks. We present *JSI Sound*, one of the first tools of this kind, developed within the Orange data mining software. The input for the tool is a series of labelled recordings. The first step allows choosing proper filtering and segmentation of the input sound. In the next step, a series of frequency and time domain features are extracted using extensive feature extraction libraries. Once the features are extracted, the data is ready to be used with the standard ML widgets implemented in Orange. Up to date, we evaluated the tool on five different domains: emotion recognition from speech, chronic heart failure detection form heart sounds, and species classifications from biosounds including bee, bird, and frog species. The evaluation results show high performance on all five domains.

1 Introduction

Artificial intelligence and machine-learning methods are slowly shifting from predefined symbols and features as input to real world signals, e.g., sounds and videos. A tool that enables fast prototyping of ML models using sound as input can significantly reduce the experimenting time both for experts and non-experts. Such tool can easily build baseline ML models for any sound-related ML task. The baseline models can be later used for comparison with more advanced sound ML approaches.

In zoology and biodiversity studies, an important task is to classify an individual animal from sound. In psychology and medicine, an important task is to recognize human emotions due to emotional disorders (e.g., depression and bipolar disorders). In medicine, an important task is to detect chronic-heart failure (CHF) which is a global pandemic currently affecting over 26 million of patients worldwide [Ambrosy et al., 2014]. Often, experts in the fields can deal with these tasks based solely on sound. However, in many situations experts may not be available. In addition, in studies dealing with biodiversity monitoring, researchers are often faced with vast amounts of data. This calls for introduction of automatic classification methods to help the experts with their work. Furthermore, non-expert may find solutions based on sound valuable when proper quality is accessed.

In this paper, we focus on the sound. Mammals, birds, amphibians, some groups of insects (such as cicadas or crickets) produce (typically structured) sounds to communicate. Aide et al. [2013] worked on a real-time acoustic monitoring system for sound-producing animals from Puerto Rico and Costa Rica. In our previous work [Gradišek et al., 2017], we used a combination of ML approaches to classify bumblebees based on the unstructured flight buzzing sound produced by movement of their wings. Ganchev and Potamitis [2007] used a score-level fusion of classifiers with non-parametric (probabilistic neural network) and parametric (Gaussian mixture models) estimation of the probability density function for identification of a series of singing insects, namely crickets, cicadas, and katydids. They achieved 90 % accuracy for 307 species. Stowell and Plumbley [2014] used unsupervised feature learning to classify bird songs. Also, for bird songs, Cheng et al. [2012] tested several machine-learning methods for classification.

Apart from speech and other forms of vocalization, human body produces sounds such as breathing sound and sound of heartbeats. A healthy heart produces different sounds compared to an unhealthy heart. Depressed human talks differently compared to a joyful one. Cao et al. [2015] used speech analysis and ML for emotion recognition. Stassen et al. [1991] analyzed speech characteristics in depressed people. Gjoreski et al. [2017] used sound analysis and a stack of ML classifiers to detect CHF.

Similar to the progress from ML systems to ML toolkits, in sound recognition tasks it would be reasonable to create a toolkit that would allow everyone with some basic computer knowledge and sound data to experiment with ML techniques - to see which filtering and segmentation methods, sound features, ML algorithms, etc., work best for a particular dataset. Here, we present our ML tool in Orange [Demšar et al., 2013; Demšar et al., 2004] that enables performing ML experiments on sound data.

The tool encompasses five filtering methods, two segmentation methods, and extraction of six types of sound features in frequency and time domain. After the feature extraction, the tool enables the use of the standard ML widgets implemented in Orange, including all of its data-manipulating widgets, feature-scoring/selection widgets, visualization widgets, ML algorithms, and evaluation widgets. We evaluate the tool on five different domains: emotion recognition from speech, CHF detection form heart sounds, and species classifications from biosounds, including bee (species of North American bumblebees), bird (species from the warbler family), and frog species (Slovenian frogs).

2 JSI Sound - Method

The overall methodology is presented in Fig. 1. It consists of five steps: input sounds, preprocess sound, extract features from the preprocessed sound, build ML models, and evaluate models. The input consists of a series of labelled sound recordings of unspecified length. It is desired that all the recordings are acquired using the same sampling rate, although this can be later adjusted in preprocessing. In the preprocessing step, five filtering methods (Finite impulse response - FIR, Butterworth, Chebyshev, Elliptic, and Bessel filter) can be used for noise reduction. In the segmentation step, the filtered signal is split into segments using sliding-window technique. During the split, the method keeps track about which segment belongs to which recording. The segmentation step is important since the original recordings are typically long, which can be computationally demanding later in the feature extraction phase. In addition, it is advised that the segments that are low in information, i.e., the energy of the audio signal in the segment is lower than the median energy in the recording, are discarded (see [Han et al., 2014] paper on speech processing). The segmentation step results in sound segments of equal length, which are then used for feature extraction. Two open-source feature extraction libraries are currently used, OpenSmile [Eyben et al., 2010] and pyAudioAnalysis[1]. These libraries can extract a huge number of features (e.g. over 1000 in OpenSmile) in time or frequency domain. OpenSmile already contains predefined features for emotion recognition from speech. Similarly, mel-frequency cepstrum coefficients (MFCC), Perceptual Linear Prediction (PLP) coefficients, and Chroma coefficients can be extracted, which all proved useful in sound processing, especially when dealing with human speech and music processing in general [Ittichaichareon et al., 2012; Yucesoy and Nabiyev, 2014]. Once the features are extracted, the data is ready to be analyzed with the standard ML widgets implemented in Orange. A typical approach would be to split the dataset into training and testing sets, to run classification algorithms (e.g., J48 decision trees, random forest, SVM, naiveBayes, and other) on the training set, to evaluate the classification models on the testing set, and to compare their performance.

[1] https://github.com/tyiannak/pyAudioAnalysis

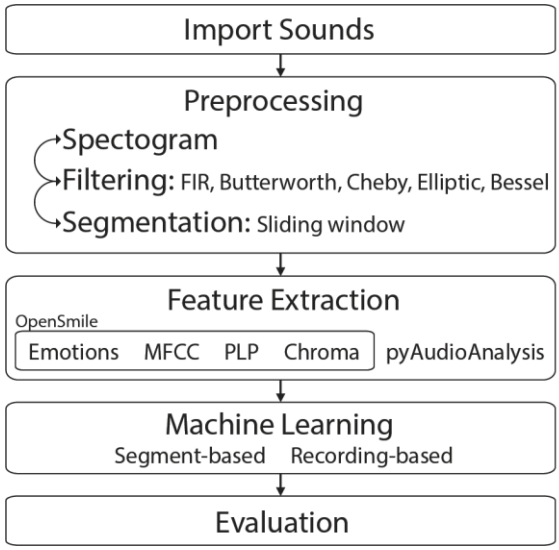

Fig. 1 JSI Sound - Method

Since the method keeps track about which segment belongs to which recording, it enables building segment-based or recording based models.

3 JSI Sound - Toolkit in Orange

Orange is an open-source ML and data mining suite for data analysis through Python scripting and visual programming in Orange Canvas. For researchers in ML, Orange offers scripting to easily prototype new algorithms and experimental procedures. For explorative data analysis, it provides a visual programming framework with emphasis on interactions and creative combinations of visual components. The Orange suits allow developing widgets by third parties, thus we implemented our methodology as Orange widgets.

After the installation, the widget appears in the list of standard widgets as "JSI Sound". In JSI Sound, the widget **Import Sounds** allows to input sound recordings from a standard file browser to Orange. Upon successful loading of the recordings, the user can use the functions offered by the JSI Sound toolkit widgets, i.e., filtering, segmentation, and feature extraction. From the filtering options, FIR, Butterworth, Chebyshev, Elliptic, and Bessel filters are available, together with parameter settings for each filter. From the segmentation options, sliding-window technique with parameters (window size and step overlap) can be chosen. After the segmentation, one can also use the filtering option to filter each segment separately. However, it is recommended that this is done before the segmentation stage due to computational complexity. The last option is the feature extraction option, which calculates features from preprocessed signal (or individual segments). The user can choose from five different feature-extraction options (Emotions, MFCC, PLP, Chroma, and pyAudioAnalysis). After complete pre-processing (filtering, segmentation) and feature extraction, the user can exploit the standard ML widgets implemented in Orange, including all of its data-

manipulating widgets, feature-scoring/selection, visualization, ML algorithms, and evaluation widgets.

Figure 2 shows how to use JSI Sound widgets in Orange. First, the audio recordings are split on two subsets (Train : Test, 70% : 30%) and are input via the sound input widget (named as Train and Test in Figure 2). Next, the recordings are filtered and segmented using the **Filtering** and **Segmentation** widgets. The parameters are chosen by checking the spectrogram of several recordings using the **Spectrogram** widget. Next, ML models (Logistic Regression, Random Forest, and Decision Tree) are trained on the training data and evaluated on the test data using the standard Orange widget, **Predictions**. The output of the **Predictions** widget is used to check the predictions of each model for the test data. These predictions are predictions from each segment of the input recordings, meaning if one recording has been split on ten segments in the segmentation phase, there will be ten predictions for the recording per model. The widget **Min/Avg/Max** allows for merging of the predictions per recording and training a meta-classifierto produce the final output for each recording.

4 Experiments

Here, we present the classification results on five diverse datasets: three datasets for animal species recognition (Birds, Frogs, and Bumblebees), one datasets for emotion recognition, and one dataset for CHF. In all domains, the recordings are segmented using a 1 s window and MFCC feature extractor, where we obtained 39 features.

The details for each dataset are presented in Table 1. The table shows number of different labels per dataset, the number of different recordings and the number of segments per dataset produced after the segmentation phase.

The experiments involve three stages: segment-based ML stage, recording-based feature-extraction stage, and recording-based ML stage. In the segment-based ML stage, six ML models are build using the algorithms: Logistic Regression, Naive Bayes, KNN, Decision Tree, Random Forest, and AdaBoost. The idea behind combining a variety of algorithms is that different algorithms can model different structures in the data. Each of the six ML models takes as input the segment-based feature vectors and outputs a probability for a segment for each class.

Table 1. Number of classes, recordings and segments for each of the datasets.

	CHF	Bees	Birds	Frogs	Emotions
# Labels	2	9	6	13	7
Recordings	152	51	81	39	535
Segments	8593	3854	5336	4447	2729

In the recording-based feature-extraction stage, the output of the segment-based ML models is aggregated and provided as the input for the recording-based ML stage. The aggregation is performed using min, max, and average over the predictions of the segment-based ML models.

In the recording-based ML stage, a recording-based ML model is trained, which produces min, max, and average probability for the segments in one recording from test set, for each class from six different ML models. These ML models (meta-classifiers) are evaluated using 10-fold cross validation on the test data. The results are presented in Table 2. The high AUC on the CHF dataset is just a confirmation from our previous study Gjoreski et al. [2017] that this is a simple ML task. The high AUC on the Frogs dataset is a consequence of a small number of individuals per class (2-3). The experimental results on Frogs and Emotions datasets are a confirmation of the results already achieved in our previous studies [Gjoreski et. al., 2014; Gradišek et al.].

Table 2. AUC for each experimental domain using 10-fold cross-validation.

	CHF	Bees	Birds	Frogs	Emotions
LR	**98**	80	**94**	**100**	**88**
Knn	**98**	75	92	99	81
RF	**98**	**81**	**94**	**100**	84
DT	96	66	83	90	69
NB	95	74	89	99	85
AdaBoost	91	67	85	93	68

5 Conclusion

We present JSI Sound - a toolkit that was developed to facilitate the sound ML tasks for users without extensive ML and sound processing knowledge. JSI Sound encom-

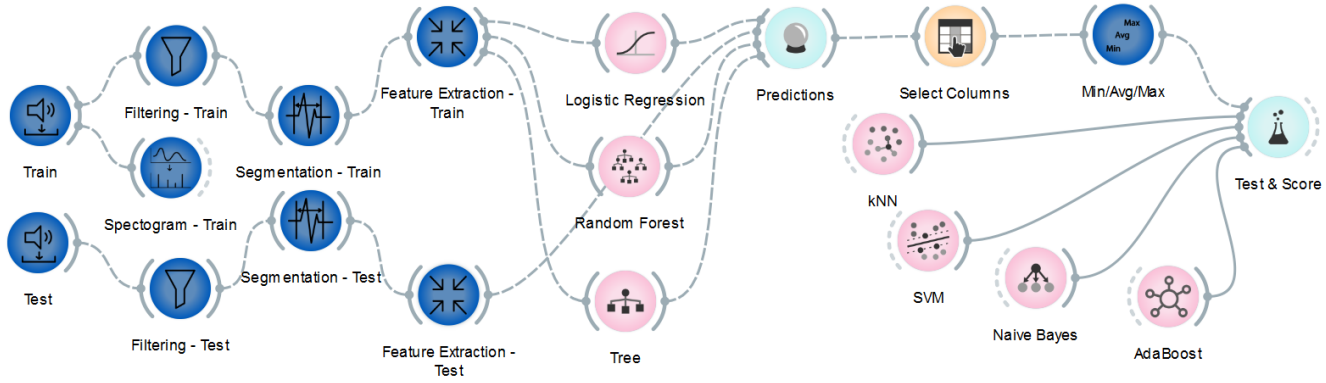

Fig. 2 Example use-case of the implemented widgets. Blue widgets are implemented in JSI Sound, others are from Orange toolkit.

passes five filtering methods, two segmentation methods, and extraction of six types of sound features, including frequency and time domain. All of them are implemented as Orange widgets.

We tested the tool on five diverse datasets – three coming from animal sounds and two from human sounds. The experimental results presented here are of high accuracy, some of them reproduced the experimental results from previous studies, which proves the suitability of JSI Sound for sound ML tasks.

Acknowledgments

We thank dr. Tomi Trilar for recordings of birds and frogs, and prof. Candace Galen and dr. Nicole Miller-Struttmann for recordings of bumblebees.

References

[Aide et al., 2013] Aide, T Mitchell, Carlos Corrada-Bravo, Marconi Campos-Cerqueira, Carlos Milan, Giovany Vega, and Rafael Alvarez. 2013. 'Real-time bioacoustics monitoring and automated species identification', *PeerJ*, 1: e103.

[Ambrosy et al., 2014] Ambrosy, Andrew P, Gregg C Fonarow, Javed Butler, Ovidiu Chioncel, Stephen J Greene, Muthiah Vaduganathan, Savina Nodari, Carolyn SP Lam, Naoki Sato, and Ami N Shah. 2014. 'The global health and economic burden of hospitalizations for heart failure: lessons learned from hospitalized heart failure registries', *Journal of the American College of Cardiology*, 63: 1123-33.

[Cao et al., 2015] Cao, Houwei, Ragini Verma, and Ani Nenkova. 2015. 'Speaker-sensitive emotion recognition via ranking: Studies on acted and spontaneous speech', *Computer speech & language*, 29: 186-202.

[Cheng et al., 2012] Cheng, Jinkui, Bengui Xie, Congtian Lin, and Liqiang Ji. 2012. 'A comparative study in birds: call-type-independent species and individual recognition using four machine-learning methods and two acoustic features', *Bioacoustics*, 21: 157-71.

[Demšar et al., 2013] Demšar, Janez, Tomaz Curk, Aleš Erjavec, Črt Gorup, Tomaž Hočevar, Mitar Milutinovič, Martin Možina, Matija Polajnar, Marko Toplak, and Anže Starič. 2013. 'Orange: data mining toolbox in Python', *Journal of Machine Learning Research*, 14: 2349-53.

[Demšar et al., 2004] Demšar, Janez, Blaž Zupan, Gregor Leban, and Tomaz Curk. 2004. "Orange: From experimental machine learning to interactive data mining." In *European Conference on Principles of Data Mining and Knowledge Discovery*, 537-39. Springer.

[Eyben et al., 2010] Eyben, Florian, Martin Wöllmer, and Björn Schuller. 2010. "Opensmile: the munich versatile and fast open-source audio feature extractor." In *Proceedings of the 18th ACM international conference on Multimedia*, 1459-62. ACM.

[Ganchev and Potamitis, 2007] Ganchev, Todor, and Ilyas Potamitis. 2007. 'Automatic acoustic identification of singing insects', *Bioacoustics*, 16: 281-328.

[Gjoreski et al., 2014] Gjoreski, Martin, Hristijan Gjoreski, and Andrea Kulakov. 2014. 'Machine learning approach for emotion recognition in speech', *Informatica*, 38: 377.

[Gjoreski et al., 2017] Gjoreski, Martin, Monika Simjanoska, Anton Gradišek, Ana Peterlin, Gregor Poglajen, and Matjaž Gams. 2017. "Chronic Heart Failure Detection from Heart Sounds Using a Stack of Machine-Learning Classifiers." In *Intelligent Environments (IE), 2017 13th International Conference on*, IEEE.

[Gradišek et al.] Gradišek, Anton, Gašper Slapničar, Jure Šorn, Boštjan Kaluža, Mitja Luštrek, Matjaž Gams, He Hui, Tomi Trilar, and Janez Grad. 'How to recognize animal species based on sound–a case study on bumblebees, birds, and frogs', *Intelligent systems : proceedings of the 18th International Multiconference Information Society - IS 2015, October 7th, 2015, Ljubljana, Slovenia*, 38-41.

[Gradišek et al., 2017] Gradišek, Anton, Gašper Slapničar, Jure Šorn, Mitja Luštrek, Matjaž Gams, and Janez Grad. 2017. 'Predicting species identity of bumblebees through analysis of flight buzzing sounds', *Bioacoustics*, 26: 63-76.

[Han et al., 2014] Han, Kun, Dong Yu, and Ivan Tashev. 2014. "Speech emotion recognition using deep neural network and extreme learning machine." In *Interspeech*, 223-27.

[Ittichaichareon et al., 2012] Ittichaichareon, Chadawan, Siwat Suksri, and Thaweesak Yingthawornsuk. 2012. "Speech recognition using MFCC." In *International Conference on Computer Graphics, Simulation and Modeling (ICGSM'2012) July*, 28-29.

[Stassen et al., 1991] Stassen, HH, G Bomben, and E Günther. 1991. 'Speech characteristics in depression', *Psychopathology*, 24: 88-105.

[Stowell and Plumbey, 2014] Stowell, Dan, and Mark D Plumbley. 2014. 'Automatic large-scale classification of bird sounds is strongly improved by unsupervised feature learning', *PeerJ*, 2: e488.

[Yucesoy and Nabiyev, 2014] Yucesoy, Ergun, and Vasif V Nabiyev. 2014. "Comparison of MFCC, LPCC and PLP features for the determination of a speaker's gender." In *Signal Processing and Communications Applications Conference (SIU), 2014 22nd*, 321-24. IEEE.