PAMBOX: A Python auditory modeling toolbox

Alexandre Chabot-Leclerc

Centre for Applied Hearing Research, Technical University of Denmark

@AlexChabotL pambox@alexchabot.net http://pambox.org

Goals

- Collect published models of auditory processing.
- Simplify comparisons across models by standardizing the interface for models that perform similar tasks.
- Make components reusable and easy to inspect and modify.

Problems and inspiration

- Intelligibility prediction work often involves comparisons across models, but the source code is rarely available. Additionally, reference levels and interfaces are often completely different.
- `scikit-learn`'s approach of providing a *consistent* interface for all methods is a strong inspiration.
- The Auditory Modeling Toolbox (Søndergaard and Majdak, 2013) is a similar attempt at collecting published auditory models, made for Matlab. Although component are well tested, they are hard to reuse.

PAMBOX's structure

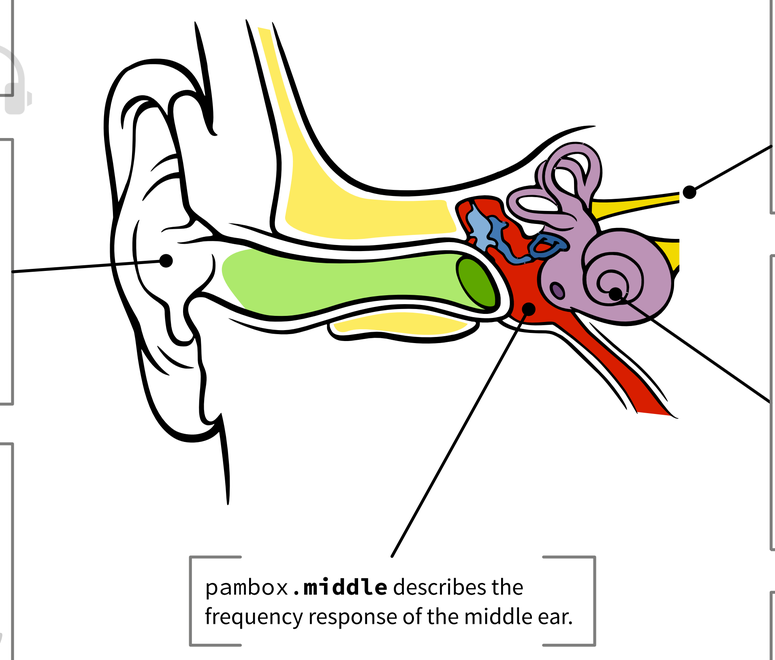
`pambox.audio` is a thin wrapper around `pyaudio` to simplify the playback of numpy arrays.

`pambox.outer` contains classes and functions related to the outer ear, such as:

- Head-related transfer functions (HRTFs);
- Headphone + ear transfer functions.

`pambox.utils` provides useful function for manipulating signals, and converting data, e.g.:

- set or measure a signal's level;
- add two signals of different lengths;
- apply filtering via FFT.



`pambox.middle` describes the frequency response of the middle ear.

`pambox.central` provides model stages from the central auditory pathway.

- Decision metrics;
- Modulation filtering;
- Ideal observer;
- Optimal detector.

`pambox.inner` regroups inner ear functions, such as:

- Cochlear filtering;
- Inner hair cell envelope extraction;
- Adaptation;
- And other filter banks.

`pambox.speech` contains speech intelligibility prediction models as well as "helper" functions to simplify experiments and comparisons across models (see section below).

Planned modules

`pambox.mono`: will regroup published models making complete predictions of monaural percepts, such as:

- Gap detection threshold;
- Modulation threshold;
- Masking;
- Signal detection.

`pambox.binaural` will contains models predicting binaural percepts, such as:

- Interaural level and time differences (ILDs and ITDs);
- Source localization.

Predicting speech intelligibility

Predicting speech intelligibility consists of predicting how well speech is understood in a given situation. Intelligibility experiments tend to have similar components, as illustrated below.

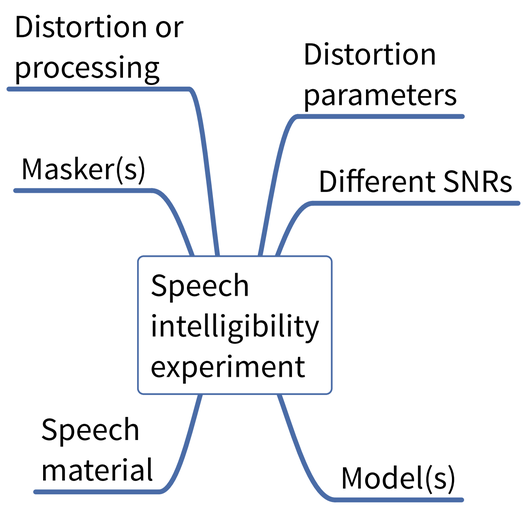


Fig. 1: Elements of a speech intelligibility prediction experiment.

`pambox` provides an interface to run experiments with an arbitrary number of models, SNRs, and distortion parameters. Currently, each experiment is treated as having a single speech material and a single "distortion", although that could be extended.

The preprocessing stage can easily be modified or replaced by overriding the `preprocessing` method of the `Experiment` class. This allows, for example, to apply the level adjustments before the distortion, or to apply processing to both the target and the maskers.

Below is an example of a complete speech intelligibility prediction experiment. Two models are compared, in a condition with a speech-shaped noise masker.

```python
from pambox.central import IdealObs
from pambox.speech import Sepsm, MrSepsm, Material, \
    Experiment

# Define models and "experimental setup".
models = [Sepsm(), MrSepsm()]
material = Material(
    root_path='stimuli',
    path_to_sentences='sentences',
    path_to_ssn='ssn.wav')
snrs = range(-9, 4, 2)

# Run experiment, transform outputs, and plot results.
exp = Experiment(models, material, snrs)
df = exp.run()
obs = IdealObs()
df = exp.pred_to_pc(df, obs.snrenv_to_pc)
exp.plot_results(df)
```

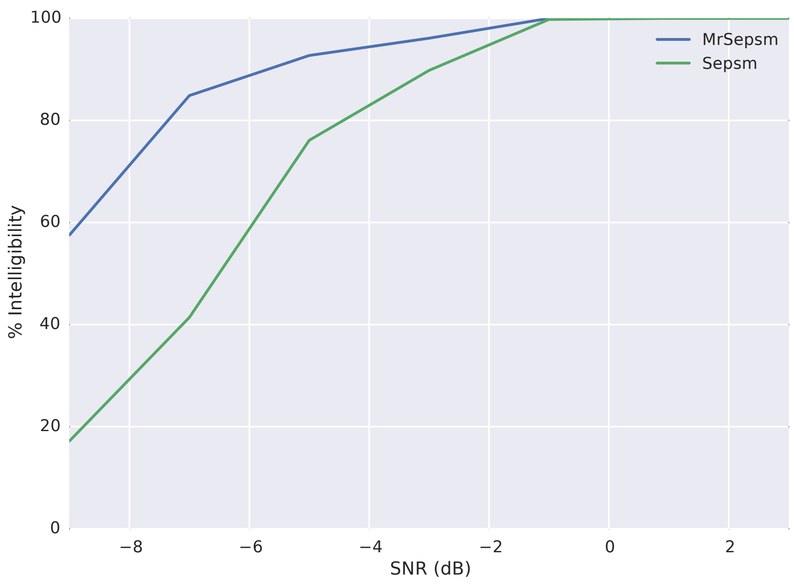


Fig. 2: Predicted speech intelligibility as a function of the SNR for two models, the sEPSM and the mr-sEPSM.

Development

- Depends on the usual suspects: `numpy`, `scipy`, `pandas` and `matplotlib` and with `pyaudio` as an optional dependency.
- Licensed under the 3-Clause BSD license.
- Docs at `http://docs.pambox.org`, thanks to ReadTheDocs!
- Hosted on Github.

Conclusions

- `pambox` is still in its infancy. Very few auditory models are currently written in Python; it would require a considerable effort to reimplement existing models.
- However, implementation of models whose code was not released would be a great help.
- `pambox` needs contributions of models stages and complete models.

`http://pambox.org`

Acknowledgements

This research was supported in part by the National Science and Engineering Research Council of Canada (NSERC), Phonak, and the Technical University of Denmark.

References

Chittka, L. and Brockmann, A. (**2005**). "Perception space—the final frontier", PLoS Biology **3**, e137, modified (Fig. 1A/Large version), vectorised by Inductiveload.

Søndergaard, P. and Majdak, P. (**2013**). "The auditory modeling toolbox", in *The Technology of Binaural Listening*, edited by J. Blauert, 33–56 (Springer, Berlin, Heidelberg).