Learning system fan and ambient noise combinations audible to PC users

Manolo Alvarez*
Department of Electrical Engineering
Stanford University
manoloac@stanford.edu

1 Background

The prominence ratio (PR) is an objective measure to assess if a tone is "prominent" or likely to be heard by the human ear [1]. It is rooted in the scientific literature that points to human hearing assessing sounds in frequency bandwidths. Tones that are within the critical band, will be heard as one. While, tones that are more than a critical band apart, will be heard separate and can mask the sound of the critical band.

If the prominence ratio exceeds 9 dB at 1 kHz or higher, the tone is likely to be heard. The PR is defined as the decibel value of the ratio of the critical band power including the tonal component and the average of the adjacent critical band power on both sides.

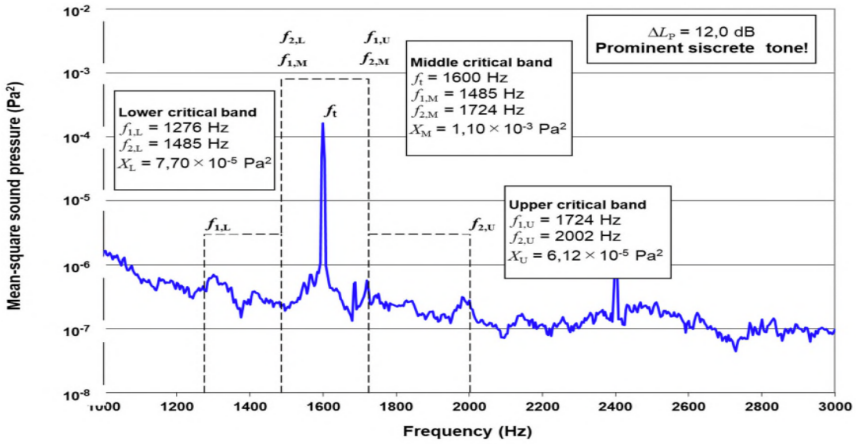
$$PR = 10 \cdot \log(\frac{W_M}{(W_L + W_U)/2})(dB)$$

where,
W_M : power of middle critical band (Pa^2)
W_L : power of lower critical band (Pa^2)
W_U : power of upper critical band (Pa^2)

For Example:

*https://profiles.stanford.edu/manolo-alvarez.

,

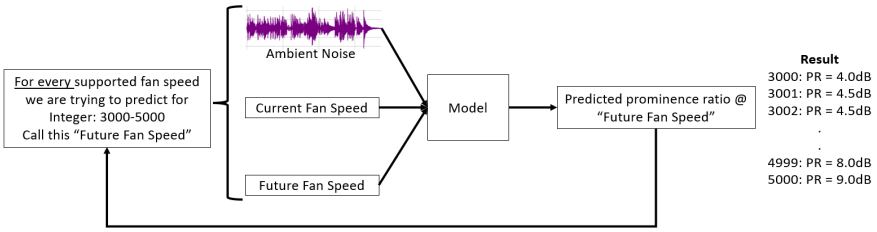


[2]

2 Introduction

The goal here is to create a model that can learn the influence fan speeds have on an environment's noise and to identify and ignore spontaneous sound.

In practice, a software would feed the model a short recording from the system microphone and the current and future fan speeds to predict the prominence ratio of each of the fan's harmonics. That would be one iteration. The software would run the model for every possible "future" fan speed and use that to find and change to the highest predicted inaudible fan speed. In other words, the model would predict how audible every other fan speed (RPM) the system could change to would be. See illustration below.



3 Motivation

Operating power in PC's is a function of thermal and acoustic constraints, and the higher it is, the faster a system will run. Most Original Equipment Manufacturer's (OEM's) set these as static constraints defined by user-configurable operating modes like "Cool," "Quiet," "Balanced," and "Performance," modes.

Static thermal and acoustic operating modes over-constrain system power levels in environments where such parameters have a negligible effect on customer experience. Sone (noise) limits are intended to prevent users from experiencing uncomfortable fan noise in the most "common" environment but the distribution of noise-level across environments is a wide one and this static limit does not take opportunistic advantage of environmental factors that boost system performance at the expense of unnoticeable fan noise to the customer.

4 Dataset

The raw data consist of 4757 2, 4, 8, and 10-second, quad-channel recordings captured at a sample rate of 48kHz in coffee shops (16.3%), offices (4.3%), homes (75.7%), and outdoors (3.8%). At first, the distribution of samples across locations was somewhat even, but towards the end of the project, I collected 2860 samples in my home in an effort to train and compare the performance on the model

with a silent and mixed set of noisy backgrounds. The results detailed later in this document will specify the representation of each location in the training set.

Over time, it was found that certain features of the raw data were redundant or useless for the model, so to simplify the model and speed up training, the following transformations were made to the data set:

1. Longer clips did not improve the accuracy of the model so every clip was trimmed down to 2 or 3 seconds.

2. Channel 0's waveform differed from channel 1,2, and 3's waveform. We, the audio expert and myself, believed that channel 0's waveform could be going through some special hardware or software filtering hidden to us. With channel 0's reliability in doubt and channel's 2 and 3 redundant of channel 1, we agreed to only use channel 1 for training.

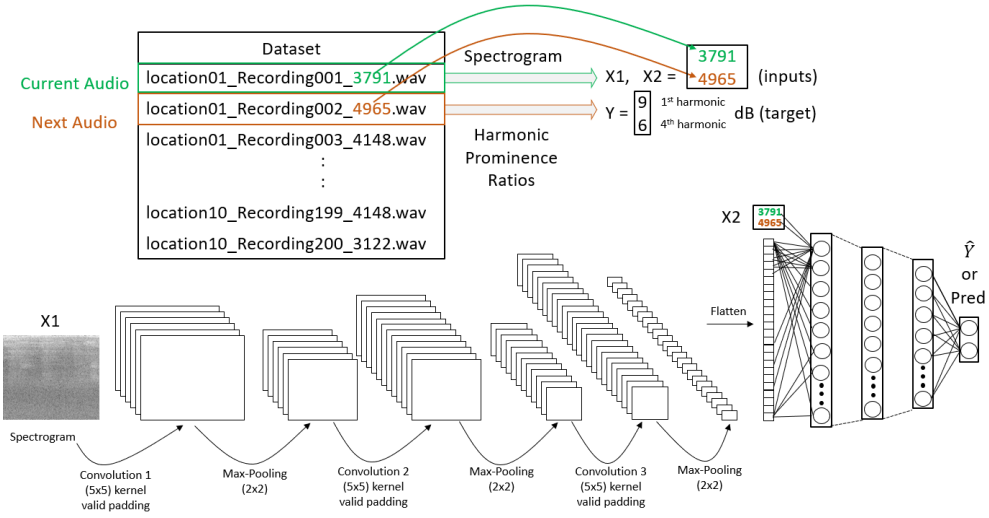
3. After placing the outlet of the fan next to the microphone blowing at full speed, it was found that only the 1st and 4th harmonics of the sound from the blades were prominent and scaled with the fan's speed. It made sense then to exclude the other harmonics from the parameters that the model was asked to learn.

4. Finally, the resulting waveform of the transformations above was transformed into a spectrogram with nfft of 512 and/or 1024.

5 Method

1. Loading the data - Purposefully, each sample of audio was saved with a filename containing the location, number in the sequence, and speed of the fan (rpm). For example, files 'location10_Recording0010_4893.wav' and 'location10_Recording0011_3509.wav', correspond to the 10th sample in location 10 with rpm of 4893 and the 11th sample in location 10 with rpm of 3509, accordingly.

2. Setting-up and running the model - I found a significant amount of online NLP projects that attained a decent level of performance running spectrograms through a CNN-based model. With those being the closest tasks I could find to mine and having tons of literature on them, I figured CNN-based architectures were a good place to start. As a result, most of my experiments were done with the CNN-based architecture below (not to scale).



3. Loss Function - It was clear early on that the model was not doing well with an L2 or MSE loss. After a careful error analysis and visualization of the data, I came to realize that the data had a few errors turned outliers that were getting blown out of proportion with MSE. Naturally, I then tried L1 and Huber losses, finding the Huber loss to be the most reasonable for my task and goals; a smaller penalty for values within 1dB and normal L1 or MAE penalty for values greater than 1dB. The Huber loss can be defined as:

$$l_n = \begin{cases} 0.5(x_n - y_n)^2, & \text{if } |x_n - y_n| < delta \\ delta * (|x_n - y_n| - 0.5 * delta), & \text{otherwise} \end{cases}$$

4. Hyper-parameter Tuning - The initial focus of the project was around the architecture, the loss function, and the best transformations for the model. Each experiment after that was part of a careful search for the best hyper-parameters with the structure laid out before. In order of priority: the learning rate, the batch size, the # of hidden units, and the # of layers were the hyper-parameters I found worth tuning based on what I had learned in class and the time I could spare. The next section will illustrate the results of each experiment with some of the parameters and hyper-parameters attempted.

6 Results & Analysis

The results are not as good as I had hoped but they are promising. Keep in mind that there does not exist a model today that can model the acoustic response of a system in a particular environment so it is my strong belief that any accuracy around 1 dB with a small standard deviation would be extremely useful. While the model isn't there, it is close, and with time, I intend to improve it.

The matrix below does not encompass all the experiments that were tried and tested. It is merely a subset of the experiments for which I have graphs for and I found to be interesting. The experiments with the yellow highlights are the ones with the best results in the mix.

Dataset (Location: Home, Office, etc.)	Loss Function	Conv. Layers	FC Layers	Learning Rate	Hidden Units	Dev. Loss
Silent - 2860 home samples	Huber d=1	3	3	1.0E-04	200	1.778
Silent - 2860 home samples	Huber d=1	3	3	1.0E-05	200	1.792
Silent - 2860 home samples	Huber d=1	3	3	1.0E-06	200	1.795
Silent - 2860 home samples	Huber d=2	3	3	1.0E-04	200	2.895
Silent - 2860 home samples	Huber d=2	3	3	1.0E-05	200	2.826
Silent - 2860 home samples	MAE	3	3	1.0E-04	200	2.919
Silent - 2860 home samples	MAE	3	3	1.0E-05	200	2.193
Silent - 2860 home samples	MSE	3	4	1.0E-04	350	16.093
Silent - 2860 home samples	MSE	3	4	1.0E-05	350	15.592
Silent - 2860 home samples	MSE	3	3	1.0E-05	200	15.527
Silent - 2860 home samples	MSE	2	3	1.0E-04	200	16.093
Silent - 2860 home samples	MSE	2	3	1.0E-05	200	15.592
Silent - 2860 home samples. >1500kHz	Huber d=1	3	3	1.0E-04	200	1.724
Silent - 2860 home samples. >1500kHz	Huber d=1	3	3	1.0E-05	200	1.775
Uniformly Mixed - 1897 samples	Huber d=1	3	3	1.0E-05	200	2.382
Uniformly Mixed - 1897 samples	Huber d=1	3	3	1.0E-06	200	2.374
Uniformly Mixed - 1897 samples	Huber d=1	3	4	1.0E-05	350	2.386

Observations/Analysis:

- It is clear that the model has a harder time learning with a mixed data set. When I plotted the prominence ratios against the rpm's for the mixed and silent data sets, I found that the distribution was significantly sparser in the mixed data set. Unsure if it is because of random factors in the environment or the fidelity of the microphones

in the system. However, the accuracy in louder environments is slightly less important for it is likely that the prominence ratio will still be low with the fan running at full speed.

- As I alluded to previously, the results across the different loss functions paint an interesting picture of the loss's distribution. The MSE loss may not be the best function to optimize the model with considering the small data set, the probability of error, and how outliers are not as important for this application. Now, looking at the model's performance with MAE and Huber, it is clear that Huber's smaller penalty for predictions within 1dB result in a lower loss, as expected. Which one is better? Hard to say but because anything within 1dB is good enough for us, I believe Huber is the better function for our task.

- Over all the experiments, 3 convolutional layers with 3 fully connected layers resulted in the best performance across all loss functions.

- The optimal learning rate varied from architecture to loss functions but the results were fairly close to one another. Close enough to notice a common floor and for their differences to be considered "in the noise."

- When I increased the the number of fully connected layers, I also increased the number of hidden units, an in most experiments I ran, the models with less hidden units performed just as well as those with more. Looking back on it though, I should've orthogonalized the experiments by keeping the number of hidden units the same as I changed the number of fully connected layers.

- It is important to note that the dev. loss was close to the train loss in most experiments with the huber and MAE functions. For that reason, I believe that either more data, different transformations, or more importantly, a different architecture would be the best next steps in improving the model's performance.

References

[1] "Tone-to-Noise Ratio and Prominence Ratio," Siemens DISW, 14-May-2020. [Online]. Available: https://community.sw.siemens.com/s/article/tone-to-noise-ratio-and-prominence-ratio. [Accessed: 12-Nov-2022].

[2] *Measurement of Airborne Noise emitted by Information Technology and Telecommunications Equipment*, ECMA-74, December 2021.