ALGORITHM FOR IDENTIFICATION AND CLASSIFICATION OF DATASETS ASSISTED BY KNN

A PREPRINT

Marius Heinrich *
Department of Computer Science
Harvard University CS50
Cambridge, MA, United States

November 27, 2020

ABSTRACT

The tinnitus retraining therapy, is to be supported with the help of an algorithm in combination with the kNN algorithm. The neurophysiological model is now used in the training of many audiologists and has found wide application in tinnitus therapy [1]. Tinnitus retraining therapy has been heralded as a major advance in alleviating tinnitus perception [2]. The goal of the research was to reduce the loudness of the tinnitus in study participants for a short period of time so that they could learn to deal with the hearing problems more easily. The algorithm I developed helps with the patient's decision making and the kNN algorithm predicts the next frequency in each iteration.

Keywords Frequency identification · Frequency prediction

1 Introduction

Critical to the success of the therapy is consideration of the brain's marvelous ability to dynamically rewire itself anatomically and neurochemically based on incoming information. To obtain datasets, two subjects were equipped with a probe, and simultaneously we ran an algorithm that played a tone in the frequency range of 0 kHz - 25,000 kHz while waiting for the subject's input. Every two seconds, the subjects gave feedback via the button whether the particular tone was a similar tone to the one they were suffering from. This allowed the algorithm to determine the tinnitus frequency. It was clear that this could have been done manually in the form of frequency adjustments with a feedback query, but we were interested in developing as automatic a procedure as possible for later implementations. We could also have used clinical datasets from patients, but these are relatively general and not suitable for focused therapy of an individual.

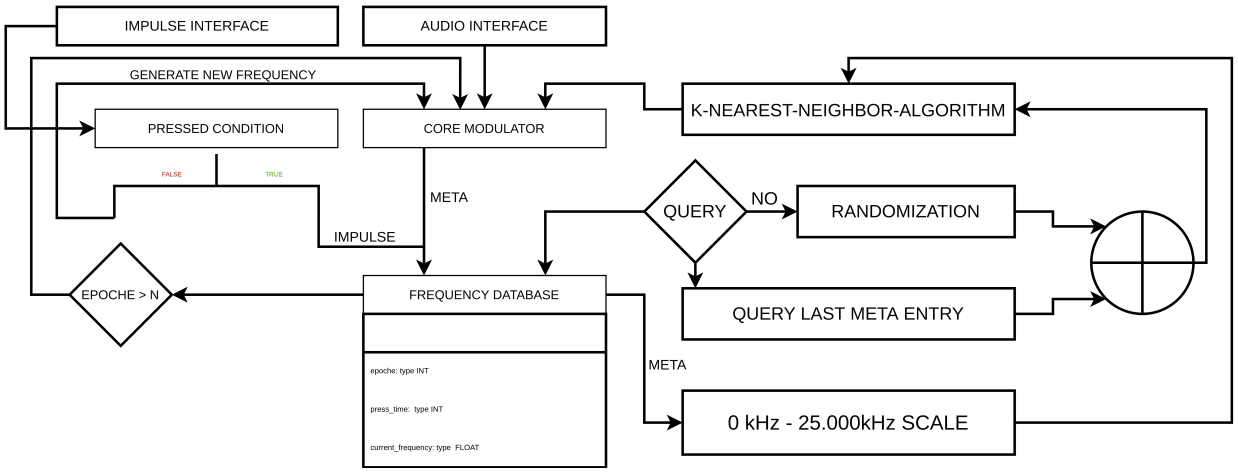
2 Forecast

The K-Nearest-Neighbor algorithm provides a real-time prediction of the next frequency in the next iteration. The computation time on the central processing unit proved to be a problem, so the computation was outsourced to the graphics unit. This speeds up the search of the K-Nearest Neighbor algorithm by a factor of 120 [3], since the NVIDIA CUDA API [3] is used here. In each iteration the algorithm approximates the tinnitus frequency, in the first iteration we use randomized data. Once started, classification and frequency finding run synchronously. The result is a dataset that can be applied to the same subject over and over again in the future. The K-Nearest-Neighbor algorithm is not modified, the Euclidean distance is calculated in the classical way in 2 dimensional space.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

*This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors

3 Algorithm

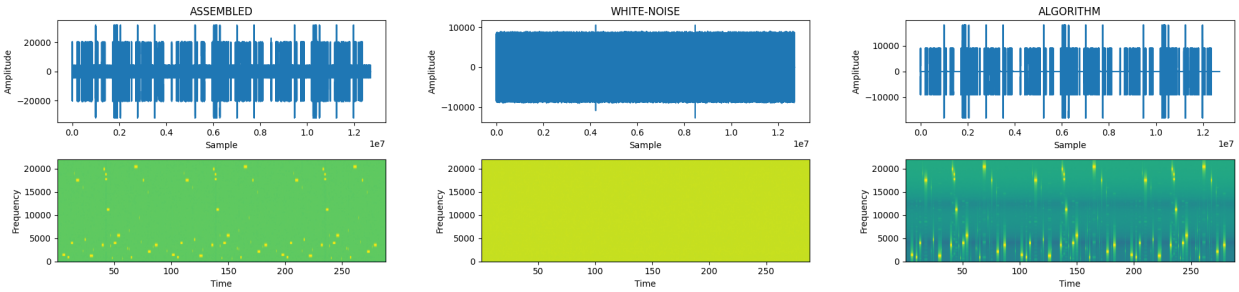


ABSTRACT ALGORITHM DESIGN OF THE APPLICATION

The algorithm includes an output and a pulse input. The input is bound to a condition that plays a central role in the classification. If the pulse input is 0, it is evaluated as false and a new frequency is initialized. This process runs recursively until a suitable frequency is found. If the pulse input receives a 1, it is evaluated as true and the pulse value is stored in a central database along with the modulator metadata. The pulse value is the result of the start time in milliseconds and the end time in milliseconds. From this, we calculate the difference and thus have the reaction time and can weigh the extent to which the frequency really fits or whether the subject was unsure. After the metadata is stored in the central database, the data runs into the scale. The scale consists of $25,000$ values intended to cover the human hearing spectrum. On the scale, we choose a radius of $(+5kHz - 5kHz)$ using our current metadata.

These values have been shown to be useful on average in the study. Along with these values, we query the database for the previous meta. If the database does not yet contain meta, we create a randomized data schema. After these steps we now have the current scale and the scale of the previous iteration, this information is now fed into the K-Nearest-Neighbor algorithm where we predict the next potential frequency. This process is also recursive and terminates after $N > epoch$.

4 Processing



With the dataset now classified and selected, we can generate an individualized therapy for the subject by embedding the frequencies suggested by the K-Nearest Neighbor algorithm into a sound image based on white noise. This can be done manually or automated in the application.

References

[1] David Baguley, Gerhard Andersson, Don Mcferran, and Laurence McKenna. *Tinnitus Retraining Therapy*, pages 168–174. 06 2013.

[2] David Baguley, Graham Beynon, and Frances Thornton. A consideration of the effect of ear canal resonance and hearing loss upon white noise generators for tinnitus retraining therapy. *The Journal of laryngology and otology*, 111:810–3, 10 1997.

[3] Vincent Garcia, Eric Debreuve, and Michel Barlaud. Fast k nearest neighbor search using GPU. *CoRR*, abs/0804.1448, 2008.