Machine Learning for Speech Error Detection

Joy Liu, 2019 Science Internship Program at Univ. of California Santa Cruz

ABSTRACT

This report summarizes my summer project at UC Santa Cruz's Science Internship Program (SIP) with fellow student Davin Nguyen, under the guidance of mentors Breanna Baltaxe-Admony (PhD candidate) and Prof. Sri Kurniawan. In this project, I have worked on building a Machine Learning method to segment speech signals to detect errors. This effort is part of SpokeIt, an interactive speech therapy software.

1 BACKGROUND

Cleft lip and cleft palate are birth defects. As the soft palate forms together from the sides of the mouth, the tissues do not fuse properly in the center, causing a split or opening in the lip and/or palate. Cleft lip and palate are fairly common, affecting 1 in 700 babies globally. Children with cleft may face many problems medical and social, like their inability to pronounce certain sounds. For instance, voices are distorted and take on a nasal quality, making speech difficult to understand.

Treatment of cleft lip/palate is a lengthy and expensive process, often involving multiple surgeries and years of speech therapy. However, many children are only able to access speech therapy periodically or not at all, and progress often lapses between appointments. To help solve this issue (currently in tandem with trained speech therapists) and provide another tool in children's development, the University of Santa Cruz's ASSIST Lab is creating SpokeIt (https://spokeitthegame.com) - an online speech therapy software for children with cleft, with current testing and training for other disabilities. The user is prompted to make a certain sound, and speech recognition technologies are used to decide whether the sound is correct or incorrect. By providing the real-time feedback, it guides the user to make adjustments towards more comprehensible speech. It offers a solution to speed up and motivate children's learning, moving them towards a more normal life - even children without consistent access to resources. In the future, it may be expanded for use as a general speech therapy tool, rather than as a cleft-palate specific one.

2 PROBLEM STATEMENT

SpokeIt's main focus is on speech error detection. For cleft speech, a few types of error are common. For instance, high pressure sounds such as "/s" are usually hard to make. A more prominent error is resonance – the soft palate may not close properly, resulting in hyper-nasality. Currently, SpokeIt's error detection mechanism works by first feeding the speech signal to a speech-to-text mechanism, then checking the resulting text for misspellings. While this is a quick-to-implement solution, it leaves some room for improvements in accuracy.

My project is part of SpokeIt's back end, focusing on speech segmentation. This problem is due to the nature of the speech signal itself. Fig. 2 shows an example recording, color-coded into different segments: pink as the signal from the therapist, blue from the child, green as background noise, and gray as silence. We want to detect error in the child's speech, skipping the therapist, background noise, and silence. This calls for a machine learning scheme to separate out the child's voice segments.

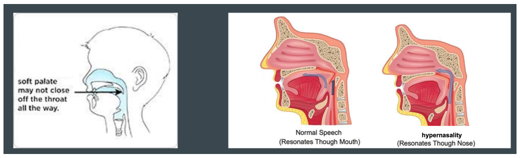

Figure 1: Resonance errors: hyper-nasality and hypo-nasality

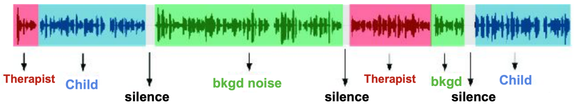

Figure 2: Speech signal (an example)

3 METHOD

For the speech segmentation problem, we worked on a data set containing 150,000 speech recordings, collected primarily from India and the US in collaboration with Smile Train, the world's largest cleft organization.

Rather than building our segmentation scheme from scratch, we would like to reuse state-of-the-art tools (e.g., python ML packages) as much as possible. To start with, we used `librosa` to label silence (the grey in the figure). Background noise is also relatively easy to separate out, given that its spectrum characteristics are very different from the human voice. The real challenge is the segmentation between therapist and child. They are both human voices, making them hard to separate – especially the difference between therapists and a correctly pronounced child. For this task, we used the python machine learning package `sklearn` and its K-means algorithm. K-means is a method to cluster samples into clusters via an iterative process:

- At the initiation stage, we choose the number of clusters, and randomly pick the cluster centroids.
- In each iteration, a sample is associated with the nearest cluster centroid.
- Each cluster recomputes its cluster centroid.

Through the iterations, samples form tighter and tighter clusters, as similar samples are grouped together. In our project, the samples are speech signal spectrum features (output of `librosa`), and the clusters correspond to therapist and child speech respectively.

We have also experimented with the deep learning package `tensorflow`. Performance is similar to that of K-means in tt sklearn, though neither was perfect.

Had we had more time, I would have continued to work on a classifier for hyper-nasality, getting to the heart of the speech error detection problem as opposed to its segmentation.