# Voice Recognition

Joy Patel, Gautam Hathi, Austin Hua

May 4, 2016

# 1 Introduction

Our project focuses on the topic of voice discrimination. While the problem of voice recognition—which involves converting voice to text—is widely discussed and explored, the problem of distinguishing between speakers—which we call voice discrimination—is also important. In particular, there are lots of situations involving new devices such as the Amazon Echo or Google Home where it is important not only to know what is being said but who is speaking. Voice discrimination can be a useful tool in these situations.

In this project, we explore a topological approach to voice discrimination. We explore different techniques for processing voice samples that can be used to create good topological features from the voice data. We then attempt to use a number of machine learning and statistical techniques to accomplish voice discrimination from topological features, and compare the results of using topological features vs non-topological features.

# 2 Data

For this project, we collected voice samples from a number of different people. We had each of these people say the phrase "open sesame" many times over (usually 50 or 60 times). This gave us a dataset with many different people saying the same phrase, and we could then proceed to try and discriminate between samples in the dataset from different people.

# 3 Process

We explored three components which we put together into a voice discrimination pipeline.

## 3.1 Audio Feature Extraction for Topology

We processed each audio sample using a couple of different feature extraction libraries. To analyze the audio sample topologically, we computed 1st dimensional homology from these features as well as from the raw audio data using the Rips complex. From the resulting persistence diagram, we took the top 10 bars as well as vectors created by binning the persistence diagram.

MFCC features appeared to give us the best homology results. The 3rd MFCC coefficient of an audio sample generally created point clouds with visible cycles in 2D PCA. When we did TDA on the point cloud, we saw clearly distinguished 1D persistence points above the diagonal:

We then used the topological features and the raw audio data as input to subsequent steps in the pipeline.

## 3.2 Distance Metrics

We explored several different distance heurstics and metrics for statistical comparisons and classification. Take note of their inputs, that is, they are metrics and heuristics between different types of features, e.g., integer valued vectors, real valued vectors, signal functions, and persistence diagrams.

1. Real Valued Vector Spaces Euclidean Metric
   $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ via $d(u, v) = \sqrt{\sum_{i=1}^{n}(u_i - v_i)^2}$
2. Integer Valued Vector Spaces Canberra Metric
   $d : \mathbb{Z}^n \times \mathbb{Z}^n \to \mathbb{R}$ via $d(u, v) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i| + |y_i|}$
3. Integer Valued Vectors Spaces Bray-Curtis Metric
   $d : \mathbb{Z}^n \times \mathbb{Z}^n \to \mathbb{R}$ via $d(u, v) = \frac{\sum_{i=1}^{n} |x_i - y_i|}{\sum_{i=1}^{n} |x_i + y_i|}$
4. Real Valued Signal Inverse Max Cross Correlation
   $d : \mathbb{C}^1 \times \mathbb{C}^1 \to \mathbb{R}$ via $d(f, g) = \frac{1}{\max_{\tau \in (-\infty, +\infty)}(\int_{-\infty}^{+\infty} f^*(t) g(t+\tau))}$
5. Persistence Diagram Distance Multiscale Kernel
   $d : PD \times PD \to \mathbb{R}$ via $d(P, Q) = \frac{1}{8\pi\sigma} \sum_{p \in P, q \in Q} (\exp(-\frac{||p-q||^2}{8\sigma}) - \exp(-\frac{||p-\bar{q}||^2}{8\sigma}))$ where $||v||$ is the Euclidean Metric and if $q = (a, b)$, then $\bar{q} = (b, a)$.

## 3.3 Statistics and Machine Learning

To discriminate between voice samples from different people, we put the features extracted from the samples into classifiers and conducted statistical tests.

### 3.3.1 Two Sample T Test

We took sets of feature vectors from voice samples of different people, calculated distances between pairs of vectors within and pairs of vectors across sets, and ran a two-sided t-test to determine whether the distances of vector pairs within a set were statistically distinguishable from the distances of vector pairs across sets.

### 3.3.2 Classifiers

We trained classifiers on feature vectors extracted from voice samples of people and used the different distance metrics mentioned above for classification. We experimented with both binary classification (using 2 labels at a time) as well as multi-label (4 labels) classification. We use the following sets of (feature, distance metric) pairs: