VOICE RECOGNITION USING MATLAB*

David Roberts

This work is produced by OpenStax-CNX and licensed under the Creative Commons Attribution License 3.0^{\dagger}

Abstract

The following module describes the process behind implementing a voice recognition algorithm in MATLAB. The algorithm utilizes the Discrete Fourier Transform in order to compare the frequency spectra of two voices. Chebyshev's Inequality is then used to determine (with reasonable certainty) whether two voices came from the same person. All material in this module is the result of a course project for a Partial Differential Equations course (Math 480) held at California State University Northridge during the Fall 2009 semester. The project was carried out under the guidance of professors – Carol Shubin and Gloria Melara.

Voice Recognition M-Files Click here to download.¹

1 Initial Problem

A human can easily recognize a familiar voice however, getting a computer to distinguish a particular voice among others is a more difficult task. Immediately, several problems arise when trying to write a voice recognition algorithm. The majority of these difficulties are due to the fact that it is almost impossible to say a word exactly the same way on two different occasions. Some factors that continuously change in human speech are how fast the word is spoken, emphasizing different parts of the word, etc... Furthermore, suppose that a word could in fact be said the same way on different occasions, then we would still be left with another major dilemma. Namely, in order to analyze two sound files in time domain, the recordings would have to be aligned just right so that both recordings would begin at precisely the same moment.

2 How to Compare Recordings

Frequency Domain

Given the difficulties mentioned in the above paragraph, it became quite evident that any voice analysis in time domain would be extremely impractical. Instead, an analysis of the frequency spectra in a voice (which remains predominately unchanged as speech is slightly varied) turned out to be a more viable option. Converting all recordings into frequency domain (by applying the Discrete Fourier Transform) greatly simplified the process of comparing two recordings. That being said, working in frequency domain also provided a new set of issues that required attention.

Finding a Norm

Due to nature of human speech, all data pertaining to frequencies above 600Hz can safely be discarded. Therefore, once a recording is converted into frequencey domain, it could then be simply regarded as a vector

^{*}Version 1.4: Jun 3, 2010 12:56 am +0000

 $^{^{\}dagger}$ http://creativecommons.org/licenses/by/3.0/

 $^{^{1}}$ http://cnx.org/content/m33347/1.3/module_export?format=zip

in 600-dimensional Euclidean space. At this point, a comparison between two vectors could easily be carried out by normalizing the vectors (giving them length 1) then computing the norm of the difference betweeen the two (of course, the difference between two vectors in \mathbf{R}^{600} is performed by subtracting componentwise). Unfortunately, exactly which norm to use is not immediately clear. After carefully comparing and contrasting the use of the Taxicab, Euclidean, and Maximum norms, it became clear that the Euclidean norm most accurately measured the closeness between different frequency spectra. Once the norm function was chosen, all that remained was to decide exactly how small the norm of the difference of two vectors had to be in order to determine that both recordings originated from the same person.

Chebyshev's Inequality

Recall that Chebyshev's Inequality states that in particular, at least 3/4 of all measurements from the same population fall within 2 standard deviations of the mean. Hence, in response to the problem posed at the end of the previous paragraph, the following solution can be formulated:

By requiring that the norm of the difference fall within 2 standard deviations of the normal average voice, we are then ensured that at least 3/4 of the time, the algorithm would recognize a voice correctly.

3 Algorithm Instructions

All files pertaining to the algorithm are located within the zip-file VoiceRecognition. zip^2 which can be downloaded by simply pressing the link. The following is a short synopsis regarding the proper execution of the software.

Short Description

As mentioned before, all files pertaining to the project can be accessed using the link: Voice Recognition³. As soon as the file is opened, the following folders will be accessable: David's Recordings Matlab Files

The contents of these folders will now be discussed in more detail. The folder *Matlab Files* contains 10 audio recordings of David Roberts saying his name 'David'. Moreover, the folder contains the two m-files project.m and voicerec.m.

Project.m is the voice recognition algorithm that accomplishes the goals of the class project. The script file project.m can be executed by typing 'project' in the command window. Please make sure that the directory in Matlab is set to the directory that contains project.m and the 10 audio recordings g1.wav through g10.wav. Once project.m is ran in Matlab, it will then request that you "Enter the name that must be recognized". Since the recordings in that folder are of David Roberts, then type in 'David'. Next, the program will inform you that you have 2 seconds to say the name 'David'. After recording, Matlab will playback the sample and give you the option to try again or to proceed if satisfied. A plot is then generated depicting how the normalized frequency spectra in your voice (top window) compares to the average normal vector of David's Voice (bottom window). See the figure below for an example. At this point, the algorithm makes a comparison and displays in the command window 'YOU ARE NOT DAVID!!!!' if you do not fall within 2 standard deviations of the normal average voice. If you do happen to fall within 2 standard deviations, then the command window displays 'HELLO DAVID!!!'.

²http://cnx.org/content/m33347/1.3/module_export?format=zip ³http://cnx.org/content/m33347/1.3/module_export?format=zip



Figure 1: Example of a Frequency Spectra Comparison.

The second m-file in that folder is voicerec.m. This script file is executed by typing 'voicerec' in the command window. Running voicerec.m will prompt the user to record their name 10 times. The recordings are then saved as g1.wav through g10.wav in the directory. Therefore, the ten new recording will in fact replace the recordings of David Roberts. Doing this results in the conversion of project.m into a voice recognition algorithm for the user's voice (as oppose to the voice of David Roberts). In this case, the user's name should be entered as the voice to be recognized (instead of 'David') when running project.m. Lastly, since voicerec.m replaces g1.wav through g10.wav in the directory, back-up copies of David Roberts' voice are conviently stored in the folder David's Recordings.