CHAPTER 11

Voice Recognition

Rodrigo Martinez and Abhishek Verma

CONTENTS

11.1 PUBLIC DATASETS

11.1.1 CHAINS Corpus

CHAracterizing Individual Speakers (CHAINS) was used to determine the accuracy of identifying an individual by an assortment of methods. The CHAINS project is used to identify key features in voices unique to individuals, which are possibly shared between speakers who adopt each other's style of speaking. The datasets included 36 speakers (18 males and 18 females) from East Ireland, from those who speak Eastern Hiberno-English, from the United Kingdom, and the United States. All subjects are recorded under several speaking conditions. This allows speaker comparison across a variety of well-defined speech styles (http://chains.ucd.ie/ftpaccess.php).

11.1.2 MIT Mobile Device Speaker Verification Corpus

This dataset is comprised of 48 speakers; 26 males and 22 females. The process consisted of having speakers recite short phrases, names, and ice-cream flavors within 20-min sessions. There were two separate sessions, one had the 48 speakers while the other contained imposters allowing each individual speaker to have their own dedicated imposter. In doing

this, it allowed all imposter files for a speaker to be provided by the same imposter.

CASE STUDY 1

By having a microphone conveniently installed into any mobile device, the user is capable of verifying their identity in a manner more secure than having a four-digit passcode. In the article "Mobile biometrics: Joint face and voice verification for a mobile platform," by P.A. Tresadern [1], both facial recognition and voice verification are combined to investigate the possibility of having a reliable method of security. To ensure the ability to verify one's identity with one's voice, the system must be able to actively detect the vocal activities of the speaker and then verify that the speaker is in fact the owner of the mobile device, or is authorized to use the said device.

The first step in being able to validate one's identity through voice on a mobile device, comes from the ability to detect voice activity. In order to use a sample collected from the microphone, the system needs to separate the speaker's voice from the background noise (or anything that isn't the speaker). This provides the system with a clear sample without any factor that might interfere in successfully authenticating the owner. Because there are many variations in speech, be it physiological traits (lisps or accents) or other forms of vocal habits that might alter the clarity of the voice, the ability to detect the speaker's voice for the initial sample and overall verification differs between speakers. For example, someone with a cold, flu, or varied form of allergies, will have an altered voice producing varied results that could affect the outcome of the authentication process. When capturing voice, its shape can be represented as a feature vector and must be condensed into a fragment at any desired location. The summarized vector can be displayed using a technique known as *cepstral analysis* which helps calculate the spectrum using Fourier transform to breakdown the logarithm of the vector with a second Fourier transform or discrete cosine transform [1], eventually charting the results onto the *mel* scale; this scale helps to perceive the variances due to distance in pitch. The second breakdown produces *mel-frequency cepstral coefficients* (MFCCs).

To determine what classifies feature vectors as actual *speech* or *nonspeech*, the Gaussian mixture model is used to categorize these, while disregarding the ordering of the feature vectors. Using the GMM is efficient for samples with a high signal-to-noise ratio (SNR), but falls short when provided with environments with significant amounts of background noise. In these occasions, artificial neural networks (ANNs) assist in classifying the vectors as either a phoneme, the smallest unit of speech that determines the difference between words and nonspeech. According to Tresadern et al. [1], the output of the ANN produces a vector of probabilities that correspond to phonemes and nonspeech which are leveled over time using a hidden Markov model;

this accounts for the phoneme frequency studied from training data and allows the phoneme to merge into speech samples.

After removing nonspeech, or unnecessary noise, the remaining sample can be used to determine if it is a match with the authorized individual; in doing so, the system can then approve or deny access into the system. To begin with the verification of the speaker, we use the MFCC depiction to define the sound of the voice provided. Once silence is filtered out of the sample's frames created with the MFCC, a cepstral mean and variance normalization is utilized. To categorize feature vectors, joint factor analysis based on parametric Gaussian mixture model [1] is used. In this process of categorizing or classifying, various examples are gone through to generate a more client-specific model to produce a tested connection. According to Tresadern et al. [1], their results concluded with an equal error rate (EER) of 3%–4% when it came to speaker verification.

CASE STUDY 2

When it comes to testing the capability of verifying voice in noise-riddled surroundings, it tends to be in locations with controlled noises. These examples or tests become less effective when faced with arbitrary noises that occur in everyday environments. "Robust speaker recognition in noisy conditions" [2] focuses on being able to distinguish, or recognize, a speaker while immersed in noise that you would find without prior knowledge of the noise.

The approach they took for modeling noise contained two steps. To begin, various copies of an original training set, a set containing clean speech data, were produced. From there, they would be able to introduce several different types of noise to simulate corruption of the cleanliness; done by introducing white noise at varying SNRs. The formula provided in Reference 2 takes frame vectors, corrupted samples, and probability of the occurrence of the noise condition for the speaker, and in doing so should improve the robustness of the noise in the test samples. The second step is to focus solely on test cases where the sub-bands, or noise samples, are matched to identifiable noise; done by ignoring the sub-bands that failed to match. In doing this, they can define the likelihood of producing vocal samples that match and pass verification. White noise wasn't exclusively used, because this type of electronically added noise wasn't natural or arbitrarily found in real-world environments; this was remedied by using acoustic sounds to imitate more realistic noise. To do this, they fed the samples to loudspeakers, that is, one speaker was playing the white noise and the second speaker was playing the clean sample. New acoustic sets of data were recorded by making the corruption occur at varying SNR while simultaneously playing the clean data. Using several corruption sources, like pop songs, street traffic, restaurant noise, and mobile ringtones. Ming et al. [2] were able to create a table providing the results of their testing. This table describes the effectiveness of being

able to correctly identify speech when placed in these arbitrary sounds at different SNR. The BSLN-Cln from testing the clean data was able to produce an accuracy of 98.41%. The rest of the data tends to vary, becoming more accurate with higher SNR amounts, and becoming less efficient at lower levels.

To test the actual capability of verifying a speaker, Ming et al. [2] used MIT's Mobile Device Speaker Verification Corpus, a database created for speaker verification. The recordings were gathered from mobile devices and were surrounded by realistic, environmental conditions. The database itself consists of 48 speakers, 26 males and 22 females, and 40 fake speakers, 23 males and 17 females. When conducting the experiment, the subjects (both groups) were to recite a list of ice-cream flavors numerous times to perform the training sessions and actual recording sessions. The testing itself took place in two different environments, the first being an office setting which provided low levels of background noise acting as corruption, and a street intersection which provided high levels of background noise.

During the process of gathering results for the speaker verification, it was determined that the office data was not entirely clean due to abrupt noise when microphones were turned on and off, and arbitrary background noises. By training the models in narrow-band noise, mismatching was able to be reduced providing better performances; training with wide-band noise produced worse results. The results revealed that having knowledge, or familiarity, with the noise bandwidth would help improve the performance of the model by being able to correctly filter white noise that matches the said bandwidth. By performing these types of multicondition tests, adding noises at varying SNR to simulate unknown sound, Ming et al. [2] concluded that multicondition training may or may not provide improved performance, but can help improve robustness.

CASE STUDY 3

It is rather difficult to bypass mobile devices that are protected by vocal biometrics, but it isn't impossible. Johnson et al. [3] proposed using a protocol called vaulted verification, improving on it to produce vaulted voice verification to ensure a more protected experience when using vocal biometrics for mobile devices. The vaulted voice verification process starts off by gathering information from the user, enough to better identify the real user in the future, through questions and responses. These samples allow feature vectors to be formed and divided into blocks. Chaff blocks, or dummy blocks, are formed for each of the feature blocks previously made; their appearance being identical to the original feature vector (allowing only the user to distinguish between them). Once this process is completed, the system is now secured.

The process for verifying the user begins immediately once the user attempts to access the protected system. To begin, the unidentified user must declare who they are by speaking their name/identification or by manually

typing it in. Following the initially requested information, the system will ask the user for their password. Failing to provide the password will cause the system to end the process, providing no follow-up questions. If the user is able to successfully input the correct password, the series of challenge questions will begin.

The challenge questions would be sent from the software to the device in a random order, making the order different whenever a new verification attempt is made. The response request would vary depending on the type of question asked; with multiple choice questions, the user would have to recite the correct answer, for passages required to be read, the user would have to read what is asked correctly. The responses given are then processed into a model; the system then makes a decision by comparing the responses made by the unidentified user and the saved responses from the original user. The responses will be converted into a bit string, having the questions end once the stopping condition is met; this would generally be when the required amount of bits is achieved. Once the server evaluates the bit string, it will decide whether to allow access or deny it.

Johnson et al. [3] utilized the same dataset as Ming et al. [2], that being the MIT mobile device speaker verification corpus. In doing so, they were able to produce models from the data they can use to determine the effectiveness of their security. The effectiveness of the results is measured in terms of the false reject rate (FRR) and false accept rate (FAR). The FRR depicted the percentage of users who are falsely denied access to the mobile device. FAR depicts the percentage of people allowed into the system while pretending to be someone else. Where both of these rates meet is considered the EER and is only applicable once the attacker has already cracked or entered the software; before entering the initial name and password, the attacker wouldn't be able to gather any information.

In their experiments, they assumed that the system was compromised to determine the level of security vaulted voice verification provides. In small scale tests where the initial password was not compromised, the EER generated was 0%; the baseline or average EER being 11%. Due to the need to know how protected the software is without the initial password check, another model and test were done. The EER of the new tests was 8%, still lower than the general baseline of 11%. With more tests on a larger scale, they were able to produce a baseline of approximately 6% for EER. Johnson et al. [3] state that vaulted voice verification is determined on a case-by-case bases allowing it to identify between the imposters in the tests conducted.

In terms of security, the vaulted voice verification has many layers to it. The communication between the user and system occurs over a secured encryption protocol, preventing someone from eavesdropping. If the encryption on the communication is broken somehow, and the culprit attempts a "man in the middle" attack, they wouldn't receive any additional information due to the data and keys still being encrypted. Even if the attacker were to obtain the name/id and password of the user, they would still need to make

2^n correct guesses (2^{58} due to having the 58 blocks) in which case the user can always issue new answers and keys.

CASE STUDY 4

The ability to identify a person based on their voice, specifically their characteristics and mannerisms, opens up a wide variety of applications. Novakovic [4] wants to illustrate the dependability of person identification using vocal characteristics in smarts, all while using multilayer perceptron (MLP).

Speaker identification can be classified as being able to verify a speaker simply by their voice. Speaker verification is being able to decide if the speaker, or client, is who they state they are. Speaker identification can be described as 1:N where the voice is being compared against N templates. A speaker identification system would not only be able to authenticate or verify a speaker when asked, but it should also be able to do so covertly while users are speaking normally. The speaker identification system must begin with an enrollment phase; gathering recorded voices to extract features to form voice prints. In the verification phase, the sample previously recorded is compared against several voice prints to determine the best matches possible.

There are several factors that can affect the ability of the system to identify a user, be it mispronounced words or phrases, heightened emotional states, room acoustics, or even aging and illness. Because of these reasons, identifying someone can be difficult throughout the day due to the fact that at any given moment they can be affected by an arbitrary variable that will alter the speaker's voice. Background noise also affects the efficacy of identification because environmental noise could cause speakers to raise their voices to match or surpass the noise around them, also known as the Lombard effect. When collecting voice samples, it is important to have silence before and after each spoken phrase. In doing this, it allows the system to more easily determine where speech is occurring, and it allows that silence to be effortlessly cut out. In some occasions, a speaker might vary the speed at which they talk, causing some vowels to be pronounced longer or shorter than normal; a problem that can be remedied by time aligning the samples.

To compare and identify speakers, a supervised learning technique is implemented. Novakovic [4] utilizes the MLP ANN for their supervised learning technique as it can be utilized for real-world applications. MLP contains a set of inputs which represent the input layer of the network, a layer of computational nodes, and an output layer of computational nodes. A perk of MLP is that the nodes (neurons) of any layer are connected to all of the nodes of the previous layer, meaning it is fully connected. MLP's network also contains layers of neurons allowing it to learn complex nonlinear tasks by mining significant features from the input gathered. An algorithm that is used frequently with MLP is the error back-propagation algorithm, which consists of a forward pass and a backward pass. The forward pass contains synaptic weights

that are fixed, while they are adjusted according to the error-correction rule during the backward pass. During the backward pass, the synaptic weights are adjusted to make the final output relatively close to what was desired.

CHAINS utilizes speaking conditions to determine vocal properties. There are six speaking conditions: solo speech, retelling, synchronous speech, repetitive synchronous imitation, fast speech, and whispered speech. Solo speech has the speaker read the entirety of a text sample at a natural rate after they read it silently to themselves. Retelling allowed speakers to freely read the text in their own words with no time limit. Synchronous speech had speakers, in pairs, read the text together while attempting to remain in synch. Repetitive synchronous imitation required speakers to listen to a recording of the speech, attempting to match the recorded model as best they could. Fast speech had speakers read the entire text at an accelerated rate, and in whispered speech, speakers read the text aloud in a whispered tone.

To run the experiment, Novakovic [4] used CfsSubSet evaluation, chi-squared attribute evaluation, and principal component analysis. Along with MLP, which contained additional feature ranking and feature selection techniques, Novakovic [4] used test datasets for 8 and 16 speakers with 25 features. The feature ranking and feature selection techniques would be used to discard irrelevant features in any feature vector. The results showed that the classification, or identification of a speaker, was dependent on the speaking condition; having a 23% variance in accuracy. Whispered speech had the worst accuracy when it came to speech classification, retelling being the most accurate. With MLP, the results provided showed an average accuracy of 65% when it came to identifying the speaker, concluding that vocal identification can be accurate enough to work in conjunction with other personal characteristics.

11.2 CONCLUSION

With the current need for advanced forms of security, biometric protection can be very appealing. Although most of a person's body is unique to themselves, one's voice can be difficult, nearly impossible, to imitate; allowing voice biometrics to be a more secure form of protection. With the ability to authenticate one's identity through speech (allowing access to their desired content) being currently available, the fear of having one's information taken is slowly diminishing. Although the data show that voice authorization and identification aren't 100% accurate, the chance to improve is high. Because of the possible limitations to software and the tools (microphones, etc.) used currently, it may take time for advancements to occur. The possibility of being able to walk into a smart environment and have the local system identify a speaker to better customize the experience will be plausible. In the future, it will be possible to maintain

your work, or information, private and secured behind a layer of protection only your authorized voice will be able to bypass.

REFERENCES

1. Tresadern, P., T. F. Cootes, N. Poh, P. Matejka, A. Hadid, C. Levy, C. McCool, and S. Marcel. Mobile biometrics: Combined face and voice verification for a mobile platform. *IEEE Pervasive Computing*, 12(1), 79–87, 2013.
2. Ming, J., T. J. Hazen, J. R. Glass, and D. A. Reynolds. Robust speaker recognition in noisy conditions. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(5), 1711–1723, 2007.
3. Johnson, R. C., W. J. Scheirer, and T. E. Boult. Secure voice-based authentication for mobile devices: Vaulted voice verification. In *The SPIE Defense, Security + Sensing Symposium*, Baltimore, Maryland, May 2013.
4. Novakovic, J. Speaker identification in smart environments with multilayer perceptron. In *2011 19th Telecommunications Forum (TELFOR)*, Belgrade, Serbia. 2011, pp. 1418–1421.