Lie Detector

Andy Liu
Caitlyn Zheng
Minxin Gao

Introduction

This implementation intends to detect whether a person is telling truth or lie by analyzing his/her voice and heart rate. The voice is collected using the microphone on an Android device; and the heart rate is collected using Microsoft Band 2. Ultimately, we implemented a Lie Detector based on the assumption that a person's voice and heart rate change when he/she tells lies.

Training & Classifying

Data Collection #1:

- Each teammate prepares 20 questions
- Start recording after the question is asked
- No preset truth/lie

Problems Encountered After Data Collection #1:
Since every person has different voice, using different people's audito data for training would require speaker recognition. After that, we can do lie detection for that specific speaker. This shrunk the number of data to ⅓ of what we had.
There were only 93 windows in total, and the trained model gives low precision/recall (about 40%). This meant that too little data were collected to get a meaning model. We think it makes more sense to focus and train our lie detector on one person with more number of questions that require longer duration of answers.

Data Collection #2:

- Focused on one teammate (Minxin Gao)
- Additional 50 questions
- Questions are more tailored to longer response and more thinking.
- This bring the total number of window to 280.

Window:

Since each answer takes different amount time, we are expecting non-uniform input size. We divide each input into fix-size windows, and assign each window a label of either “truth” or “lie” depending on where the answer came from.

Therefore, to evaluate each answer, we have a number that evaluates confidence of output -- which is the number of truth label over the total number of truth label. If we are to use a regressor, our confident rate would average of value from all windows.

We set our window size to cover about 1 second, and this gives us 8000 audio data with 1 heart rate data for each window.

Features:

<u>Audio</u>

- Formants features
    - This is the one we used in project A3. Using np.histogram with 55 bins over 5500 range.
- Pitch contour
    - This is the one we used in project A3. Using np.histogram with 64 bins over 128
- pyAudioAnalysis
    - This is an external library for audio analysis we found online. We are interested in the feature extraction part of the library, so we just clone audioFeatureExtraction.py and ultilities.py from the library to python/LD. The List of feature we used:
        - Zero Crossing Rate
        - Energy
        - Entropy
        - Entropy of Energy
        - Spectral Centroid
        - Spectral Spread
        - Spectral Entropy
        - Spectral Flux
        - Spectral Rolloff
        - Chroma Vector
        - Chroma Deviation
- MFCC

        - We didn’t use any MFCC features in this project. We have tried delta coefficient of MFCC, and delta-delta coefficient of MFCC, but either of them resulted in better performance (precision and recall)

<u>Heart Rate</u>

- Raw data
    - We were expecting PPG data since this what we get if we use phone’s camera to collect data. The raw data from Microsoft Band 2 is already in calculated/filtered heart rate.
- Statistical Data
    - Each window of data only contains one heart rate data, so there isn’t much we can play around with. We tried standard deviation, variance, min/max, average, but they all resulted in lower precision and recall rate or no significant difference.

Machine Learning Model:

- sklearn.ensemble.RandomForestClassifier
    - We start with Random Forest Classifier because it worked really well for the speaker identification project. However, the result is disappointing in this case. After hours of Grid Search, we were stuck at about 0.53 precision rate and 0.50 recall rate.
- Pystruct
    - We make a assumption that, if a person start lying, he/she will keep lying. We wanted to take advantage of structured learning, so we used learning model from library PyStruct.
    - Specifically, we are using implementations of Support Vector Machine with Conditional Random Field from this library:
        - CRF implementation: pystruct.models.MultiClassClf
        - SVM implementations: pystruct.learners.NSlackSSVM, pystructlearns.SubgradientSSVM
    - SubgradientSSVM takes way more times to train, but it performs better than NSlackSSVM. Although at the beginning these models give good precision (0.56) and recall (0.50), the performance did not get any better after several Grid Search runs.
- sklearn.linear_model.LogisticRegression
    - As we mentioned above, we had thought about using a regression model. We picked an simple linear model because we don’t have much data, and we want to do fast Grid Search sessions.
        - Best parameter: {‘penalty’:’l2’, ‘C’: 0.5, ‘tol’:’1e-05’,’solver’:’sag’ }

        - Precision: 0.5813, Recall: 0.5667
- sklearn.tree.DecisionTreeClassifier
    - Although Decision Tree Classifier did not work well in the speaker identification project, it is a really simple model and it can work well in this case because we don't have much data. The biggest concern is outfitting the training data. Therefore, we set boundaries on 'max_leaf_nodes' and 'max_depth' when we performed Grid Search.
        - Best parameter: {'min_simple_split':2, 'max_leaf_nodes':30, 'max_depth': 6, 'min_samples_leaf': 3'}
        - Precision: 0.647, Recall: 0.6311

We ended up choosing Decision Tree Classifier, because it has the best precision and recall rate.

Why is it not working as expected:

- Microsoft Band 2:
  The heart rate update or calculation rate is lower than we expected, and this results in having 8000 audio raw data vs. 1 heart rate data in one window.
- Heart Rate:
  We ended up not getting much out of the heart rate feature. We had 140 audio features while having only 1 heart rate features, and this makes the heart rate part of this project less significant.
- Features:
  The traditional lie detector uses ECG data, but we were unable to get ECG data. Overall, we need better or more features from audio and heart rate that help analysis speaker's emotion and mental state.
- Type of Question:
  The pitch and heart rate change not only when a person is lying, but also change when personal-related/sensitive questions are asked, even if a person is telling the truth.

Android Application Side:

Connecting with Microsoft Band 2:

We encountered many problems when trying to connect the Android App with Microsoft Band. The biggest issue was dealing with the permission. Eventually, we had to browse the Microsoft Band Developer API to learn about methods regarding connecting to the band. We also had to add a button at the end of the app to allow user consent in order to be able to collect user's heart rate. Also, we wanted the app to start collecting both audio and heart rate data once the switch is turned on, so there were a few minor changes in the application layout.

Receiving and Displaying Result from Server:
We also spent a lot of time to figure out how to retrieve the result from server and display it on our app. It turns out that we had to add onMessageReceived and braodcastSpeaker to the onConnected method in Microsoft BandService class.

Conclusion

Although the application did not work very well, we learned a lot through the process of implementing a lie detector. Since every person is different, we cannot just assume that the vital status of a person will always change when telling lies. This requires a lot more research efforts beforehand in order to reach a generalizable/reasonable assumption. We also realized the importance of available equipments. Both the Android microphone and Microsoft Band are not explicit enough for our purposes. Especially the band, since the heart rate data we are getting are already pre-processed, we do not get enough information from heart rate data since lying often happens in a very short time frame (~ms). We spent hours to find feature extraction libraries that are helpful for our application, but we think we can find/implement more useful features if more time allowed. Overall, we had fun implementing this application applying all the things we have learned throughout this semester.