Social Anxiety Prediction Final Report - Team 1

Jamie Hayes
School of Engineering and Applied Sciences
University of Virginia
Charlottesville, VA 22904–4259
jah4yq@virginia.edu

Shichen Li
School of Engineering and Applied Sciences
University of Virginia
Charlottesville, VA 22904–4259
sl4bq@virginia.edu

Saran Mishra
School of Engineering and Applied Sciences
University of Virginia
Charlottesville, VA 22904–4259
sm8dm@virginia.edu

Taylor Patrick
School of Engineering and Applied Sciences
University of Virginia
Charlottesville, VA 22904–4259
tap4xc@virginia.edu

1 Problem Definition

1.1 Motivation

Individuals who have social anxiety may face many obstacles in their everyday life and interactions. Social anxiety is defined as the fear of being judged and scrutinized by others that is beyond the normal expectation in a specific social situation. It is important we know what kinds of features can predict the changes in an anxiety episode.

First we need to detect the physical and physiological patterns of a subject. Then we must determine what measured fluctuations and tendencies are of interest. Using a machine learning algorithms, we will differentiate between different episodes of social anxiety. A suitable algorithm is required to improve the accuracy.

1.2 Problem Inputs

With six study participants (4 undergraduate students and 2 graduate students), we used the Shimmer ECG and GSR sensors to collect physiological and physical data with their context (solo video watching, dyad normal talk with/without evaluation), with 4 periods (baseline, anticipatory, experience and post-events) and then extract data streams related to accelerometer signals, gyroscope signals and skin conductance signals.

By using a sliding window algorithm, we extracted the most important features as input data, with 34 features from all these signals. Also the Social Interaction Anxiety Scale (SIAS) scores were collected by the HOBBY pilot surveys.

1.3 Problem Outputs

The outputs were the episodes of social anxiety and the changes in severity score from one period to the next period.

2 Related Work

Huang et. al. [1] proposed a feasibility study about the non-invasive mobile sensing technology by means of a smartphone app to track the GPS location data in addition to brief questionnaires which

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

students completed reflecting their moods and indicating their SIAS scores. By using the cumulative staying time and the unidirectional transition frequency as the extracted features, they used correlation and significance analysis and linear regression analysis including Least Square Error (LSE) and Least Absolute Shrinkage and Selection Operator (LASSO) to investigate if these features can predict SIAS score changes. The results showed that the cumulative staying time only had weak correlation with SIAS scores, while the transition frequency was significant correlated with these scores.

Vriends et. al. [2] conducted another study about the self-focused attention (SFA) in social anxiety disorder with two phases of experiments. The first experiment consisted of two groups of people with high and low social anxiety single women on a video conversation divided into 4 groups: warm-up, positive, critical and active. The second experiment replicated the first one with a clinical sample suffering from social anxiety disorder and a control group. The statistical methods including t-tests or chi-quadrant tests. Results showed that high socially anxious people had more tendency of SFA in critical space, but less in active phase, compared to the group with low social anxiety. In addition to this, in the 2nd experiment, compared to the control groups, women diagnosed with SAD showed higher SFA in both all four phases and self-rated anxiety during the conversation.

Finally, Kotsilieris [3] made a comparative literature search for the prediction of specific types of anxiety disorders using machine learning techniques. It was concluded that support vector machine (SVM) was the most common method especially at the prediction of SAD, and Artificial Neural Networks (ANNs), Random Forest (RF) and Neural Fuzzy System (NFS) also had some good scores.

3 Proposed Idea

3.1 Project Plan

To our knowledge, from the literature reviewed above, there are few studies using data from wearable sensors to make predictions about SIAS score changes using high-leveled machine learning techniques.

The volume of collected data for this problem, both in terms of frequency and the number of platforms, necessitates the implementation of some form of data pre-processing to simplify our data analysis and model implementation. Given our data set is comprised of time series data, our plan is to implement a sliding window to extract an array of features from our data set for implementation in our chosen machine learning model. Using the data sets and labels collected previously, our team plans to explore finding the best possible algorithm for the purpose of anxiety detection/prediction.

Utilizing the methodologies identified by the literature we have reviewed and in-class material, we will do a comparison between the various kinds of ML architectures that will give us the highest prediction rates. Algorithms of interest from supporting literature include: Linear Regression (LG), Support Vector Machine Regression (SVMR), Least Absolute Shrinkage and Selection Operator (LASSO), Multi-layered Perceptron (MLP), and K-Nearest Neighbor (KNN), Gaussian Naive-Bayes (GNB), and Random Forest (RF).

Our modeling procedure followed the general framework: Selection of emotional/psychophysiological stimuli, parsing and filtering of EKG and GSR data, identifying and ranking the proper classification algorithm, and finally, mapping the stimuli with relation to the results of the classification. Further research into the proper time-series methodology also needs to be conducted. Overall, we will use a ranking system to summarize our findings.

3.2 Expected Outcome

The ideal outcome of this project would be detecting episodes of social anxiety. Furthermore, we hope to use our data features to detect changes in the severity of social anxiety episodes reflected in the score from the questionnaires given to subjects (mentioned previously in Problem Definition section). These SIAS score changes, or the label set, will be utilized to evaluate and maximize the accuracy of our implemented model.

4 Experiments

Machine learning models were explored and applied to identify features that aid in detecting episodes of social anxiety. A combination of python sklearn and the keras wrapper were utilized for model development and data processing. Data was previously separated into two second windows with one second of overlap. Thirty-four features obtained from accelerometer and gyroscope sensors were analyzed as the data set. These features include mean, std, min, max, [25, 50, 75] percentile, energy, entropy, var, kurtosis, mean absolute deviation. The Social Interaction Anxiety Scale (SIAS) Score was the label set. Timing windows (data rows) with missing or 'nan' values were removed. Training and testing data were split at a percentage of 80 and 20 respectively and implemented using sklearn.model_selection.train_test_split. Our team utilized error functions to understand and interpret the outcomes of all models trained on a large training data set. The error or loss was used to determine the amount of measurable error between the predicted outcomes based on training data and the actual target values. The loss functions used include Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

4.1 Experiment Results

4.1.1 Linear Regression (LR)

Linear regression (LR) is a simple, error-based Machine Learning model that captures the relationship of continuous features.

In this case, the dependent variable (SIAS score change) is dependent upon several independent variables (features extracted from sensors). A regression model involving multiple variables can be represented as an equation of hyperplane:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

The sklearn.linear_model.LinearRegression() function produces the LR model which was applied to each feature separately, with the SIAS score change as the label set. Linear regression was a poor performing algorithm, with the minimum error for individual features (MSE = 1.152684, MAE = 0.838860, RMSE = 1.073631, the entropy of gyroscope) and the maximum error of (MSE = 1.324269, MAE = 0.901702, RMSE = 1.150769, the entropy of accelerometer). In addition to using all these features, we still can't achieve an ideal outcome (MSE = 1.1269, MAE = 0.8550, RMSE = 1.0616)

4.1.2 Support Vector Machine (SVM)

Support Vector Machine Regression (SVMR) support standard linear and nonlinear classification and regression. The SVMR algorithm attempts to fit as many instances as possible in the +/- ϵ bound while limiting margin violations. A SVMR model with linear kernels, a trade complexity C=100, and an automatically generated gamma value was produced utilizing the sklearn.svm.SVR() function. SVMR was also a poor performing model, with the minimum error for individual features (MSE = 1.1762, MAE = 0.8233, RMSE = 1.0845). The SVMR class becomes very slow when the training set grows large, this had a limiting effect on the model training. Also, the data set may not have been significantly linearly separable for the given label set.

4.1.3 Least Absolute Shrinkage and Selection Operator (LASSO)

LASSO Regression performs a so-called L1-regularization which is a process using additional information introduced for the sake of overfitting prevention. The minimization objective here including the residual sum of squares (RSS) and the sum of absolute value of coefficients.

The formula here can be expressed like this:

$$RSS = \sum_{i=1}^{n}(y_i - (\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}))^2 + \alpha \sum_{j=1}^{p} |\beta_j|$$

Where n means the number of observations of the dataset, and p denotes the number of variables. x_{ij} represents the value of j-th variable for i-th observation. And α acts as an important parameter who is

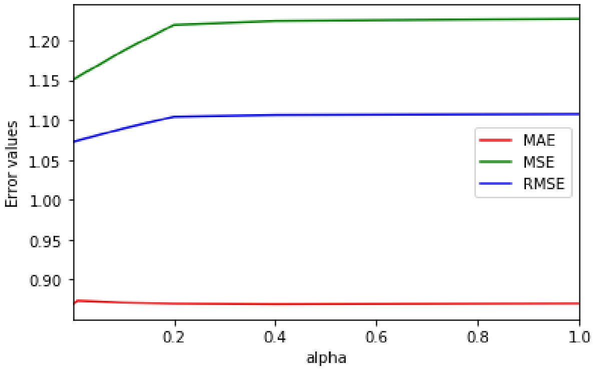

Figure 1: The Relationship between Parameter α and Error Values in LASSO Model

responsible for balancing the RSS and the sum of absolute value of coefficients. The higher value it is, the more features' coefficients will be zero. When alpha is 0, Lasso regression produces the same coefficients as a linear regression. And when alpha is large close to infinity, all coefficients are zero.

Finally we found that the smaller alpha can lead to lower MAE, MSE and RMSE rate, which is shown in Figure 1. Then we also implement this algorithm for each feature with $\alpha = 1e^{-10}$.

The results here is that the overall model yields an MSE of 1.1513, RMSE of 1.0730 and MAE of 0.8687. For each individual feature, it turns out to be like this: the entropy of gyroscope has the maximum error with MAE = 0.892304, MSE = 1.277005, and RMSE = 1.130047; the entropy of accelerometer has the minimum error with MAE = 0.835701, MSE = 1.149970, RMSE = 1.072366. This is quite contrary to the result for each individual feature, compared to LR model. But with respect to the overall performance, it is nearly equivalent to that of LR.

4.1.4 Multi Layered Perception (MLP)

Neural Networks are modeled after neurons (axons and dendrites), in the brain. Essentially, we design sets of neurons which are all connected via nodes or pathways amongst one another. For our purposes, a neuron is an object which holds a weight – a numerical value between 0 and 1. Often, this is called the activation of the neuron. Layers of these neurons are stacked together. In general, most MLP structures are comprised of three distinct types of layers: input layers, hidden layers and output layers. For our analysis we utilized two separate MLP algorithms, one with a single hidden layer and the second with three hidden layers to see if the increase in parameters make a difference in minimizing the cost function for this dataset. Though powerful, the idea is quite simple. We look at which combination of inputs form a pattern of neurons which have activated in some capacity in order to then produce an output.

MLPs or feed-forward Artificial Neural Networks are known for the fact that they can handle the problem of non-linearity in a data set. For this problem, as we decided this was a regression problem, the MLPs were designed to do that task. Since we have 34 features, we will have 34 neurons making up our input layer. Then we experimented with a single hidden layer followed by three hidden layers. Both of the layers discussed had a ReLu (Rectified Linear Unit) activation function.A ReLu, in short, is a linear activation function which gets rid of any values less than 0. Earlier we discussed about the weights of the neurons otherwise and thus, the ReLu function activates a singular neuron if and only if the value it carries is positive. Finally, the output layer has a linear activation function as this is a regression problem. Our loss function was MSE, as previously discussed and our optimizer was the ADAM optimizer.

Our initial hypothesis was given the fact that MLPs are heirarchical models, we expected them to be good performers. For a single hidden layer MLP, the MAE, MSE and RMSE are as follows: 0.11238, 0.02379, 0.1542, respectively. For the three hidden layer MLP the MAE, MSE, RMSE are: 0.07790, 0.01355, 0.1164.

Thus, we can see that the increase in parameters did have a positive effect on the mean square error for this data set as the 3 hidden layered MLP performed the best. Both MLP algorithms were run for

100 epochs and had a batch size of 250. A Backpropagation algorithm was run simultaneously with the forward propagating nature of the MLP in order to minimize the cost function i.e. finding the negative gradient of the cost function.

4.1.5 K Nearest Neighbor Regression (KNN)

We wanted to use a diverse array of regression algorithms and this lead to the utilization of the K-Nearest Neighbor approach to a regression problem. The set up is very similar to the simple KNN algorithm used very commonly for classification.

$$D\left(X,Y\right) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}$$

The general idea being that, we utilize similarity measures between input variables based on distance measurements to predict a numerical target. For our case, utilized the generalized Minkowski distance measurement for our calculations. Our results for our measurement metrics are for MAE, MSE, RMSE, the results are 0.35588, 0.6113, 0.78187, respectively.

We went one step further to tune the model. By creating a validation set from the data, we sought to find the ideal "K" and re-run the model. The ideal "K" for this data set is 5. MAE, MSE, RMSE are 0.4228, 0.5261, 0.72536, respectively.

4.1.6 Gaussian Naive-Bayes (NB)

In this project we choose Gaussian Naive Bayes, and it is based on the Bayes' theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

A common assumption here is called conditional independence, implying that the effects produced by every cause are independent among each other and therefore we can get:

$$P(y|x_1, x_2, \ldots, x_m) = \alpha P(x_1|y)P(x_2|y)\ldots P(x_n|y)P(y)$$

where α is the normalization factor, y belongs to one of the P different classes in $Y = \{y_1, y_2, \ldots, y_n\}$ and $\mathbf{x}_i = [x_1, x_2, \ldots, x_m]$ stands for a feature vector.

By using the sklearn.naive_bayes.GaussianNB() function in python, we test the error values with the random states parameter ranging from 10 to 100, and the results are shown in Figure 2.

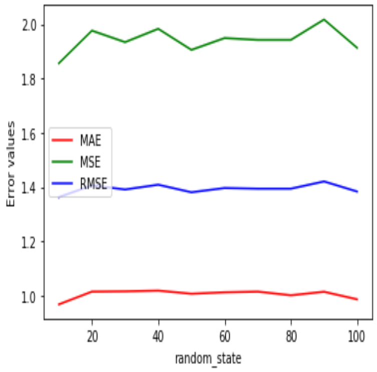

Figure 2: The Relationship between random states and Error Values in Gaussian Naive Bayes Model

From this result it is evident that when there are 10 random states, the minimum error values are achieved, but even in this condition the accuracy is very low with MAE = 0.966822, MSE = 1.856402 and RMSE = 1.362499. And the accuracy score is only 34.21%. This is the model with the worst performance.

4.1.7 Random Forest Regression (RF)

The random forest model is comprised of a series of decision tree estimators created from the training data set. The model regression output is the average output of the individual trees.

For the purpose of the experiments conducted here, several RF models were constructed at training on feature extracted data with the number of estimators n ranging from $n = [10, 300]$.

While true that the models with the greatest number of trees generally minimize error metrics the most, diminishing returns with regard to error minimization was observed after $n = 50$. Additionally, models with greater numbers of trees require additional training time. Therefore, to minimize error and training, $n = 50$ (MSE = 0.1738 , RMSE = 0.1267 , MAE = 0.3560) was found to be the optimal number of estimators for the RF regression model.

4.2 Experiment Analysis

Optimal performance is achieved by models that minimize error metrics- MSE, RMSE, and MAE. Based on this evaluation, it can be observed that MLP (3 layer) achieves the most optimal performance of all the tested models.

5 Conclusion

Our goal in this analysis was to utilize a variety of Machine Learning architectures to find the best least square estimator to validate the importance of the physiological features extracted from the prior study i.e. the features gathered utilizing the sliding window algorithm.

We wanted to analyze simpler models and progressively utilize hierarchical models such as Random Forests and even a deep learning methodology (MLP). We can see that the results do meet our expectations as the "stacked" models outperformed the simple regressors when the model was predicted upon the test data set.

Understanding the importance of the physiological features gathered, we will now be able to adequately diagnose incidents of social anxiety. And as such, there are many directions to take this research further. One thing our group remains particularly excited about is utilizing the methodologies created in this analysis to accurately predict the severity of social anxiety inducing episodes.

6 References

[1] Yu Huang, Haoyi Xiong, Kevin Leach, Yuyan Zhang, Philip Chow, Karl Fua Bethany A. Teachman, Laura E. Barnes, Assessing Social Anxiety using GPS Trajectories and Point-Of-Interest Data, Ubicomp'16, 2016, Sep.12-16, 898-903.

[2] Noortje Vriends, Yasemin Meral, Javier A. Bargas-Avila, Christina Stadler, Susan M. Bogels, How do I look? Self-focused attention during a video chat of women with social anxiety (disorder). Behaviour Research and Therapy 92 (2017) 77-86.

[3] T. Kotsilieris, E. Pintelas, I.E. Livieris and P. Pintelas, Reviewing Machine Learning Techniques for Predicting Anxiety Disorders, Technical Report, No. TR18-01.