Independent Study Report

Bharat Mukheja

1 Introduction

MetaTutor is a very good framework for interactive learning and intelligent teaching. It not only provides system for students to interact with the learning environment much more personal feedback than standardized learning environments. Every tutoring session with this system collects data from a variety of environment sensors, as well as face sensors. Theoretically speaking, these sensors provide a picture of the environment for the student at each time-step. I have analyzed the collected data below, to find out it provides enough information for a model to be able to predict what the student is going to emotionally experience next.

2 The framework and its data collection

A major chunk of time in the project was involved in understanding of data sources, the significance of different features and merging of data from different sources in the framework. The data merging is covered in the feature extraction part.

2.1 Framework

From the point of view of data collection, MetaTutor has 3 components - 1. The activity agents, which conduct the tasks student engages with primarily - Presenting the reading material on screen, conducting and guiding a student through a test/questionnaire, playing videos guiding work-flow based on student choices and feedback. The data collected by these agents is stored in 'Log Files', which I would refer by the same name going forward. This has a variety of agents, as well as descriptions of their activities with details if needed. The data is mostly tabular.

2. On Screen sensors, which sense facial muscle movement and eye positions and orientations. This is the data I have worked mostly on. This has around 26 muscle readers Action Units(AU's),Emotion Evidence scores, Emotion Intensity scores, Face Orientations, Timestamps, Accel data and more. 3. Eye tracking data form wearable glasses. These detect where the eye is currently looking at on the screen. It has eye coordinates, and more.

2.2 Evidence and Intensity Scores

FACET has defined evidence and Intensity scores in their own way. Evidence is a probability between 0 and 1 for the odds of an emotion against the absence of that emotion on a logarithmic scale. Evidence = +1 ($10^1 = 10$). The observed expression is 10 times more likely to be categorized by an expert human coder as joyful than not joyful. Intensity is the intensity score of emotion or action unit.

2.3 Feature extraction

For my analysis, I extracted the emotion Evidences, AU values, and Agent details from log files. I merged the agent details with AU's to create a predictor variable and each emotion represents a response variable. This became a multivariate classification problem.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

2.4 Parameter Selection

The probability maps provided to us don't have class labels. But the final output needs to have classes associated with each of the segment. This is an unsupervised model because we don't know how many segments we are going to get in the final result. Hence we would need to do parameter estimation by learning from the data itself. We are going to adopt Expectation Maximization for the same. EM assigns labels and estimates parameters simultaneously. For the given data, the E step of EM would first compute the probability distribution using the existent parameters. The M step would update the estimates of parameters based on weighted labelling.

2.5 Performance Metric(Model Evaluation)

Our response variables are exponential scales represented as decimal numbers. For example, the scores for 'Joy Evidence' provide integers between range -3 and 3, and they represent the presence of the emotion Joy over its complimentary emotion, which for our purposes can be called as 'Not Joy'. Hence if the value of Joy Evidence is 1, there is evidence for presence of 10 times more joy than everything except Joy. This type of data leaves us with options such as converting the exponential scale to a linear scale for linear models, or treat the data as classification labels(+ve is 'Yes' and -ve is a 'No') and use it as a classification model, and the latter makes more sense because the linear scare required for regression would span very wide and make LSTM training difficult. On the classification mode, I have used the univariate and multivariate accuracy, precision and recall wherever needed and reported metrics as such.

3 What we need?

The first motive of the exercise is to find out how well I can predict the emotional responses, which if possible to a good degree, will allow me to research more on the actual emotion and response characteristics of different influencing factors, such as the content being played on the screen and more.

For the prediction, I deployed three models to compare their performance. First was plain logistic regression which also served as a baseline model, Second was a boosted gradient tree which ought to be a non-temporal prediction model, and the third was a multilayer LSTM model. The number of layers in the lstm was not fixed till implementation to leave a room for experimentation. I worked my way up from 1 layer up to 3 layers and included the findings in my result. Essentially having more layers didn't affect the outcome too much.

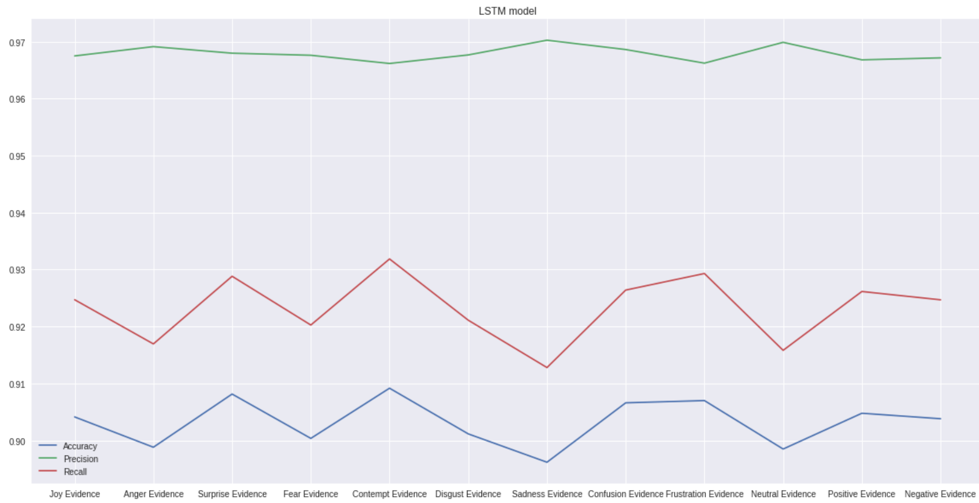
Also, in practice, non-temporal boosted trees have much higher performance over neural networks both in terms of time and accuracy, or unless the neural networks are highly optimized or very skillfully created.

4 Results

4.1 Comparison over Neutral Emotion

Percentage Values	Regression	Gradient Boosted Tree	LSTM
Accuracy	0.94	0.96	0.91
Precision	0.95	0.97	0.97
Recall	0.97	0.98	0.93

4.2 LSTM Performance over all emotions



4.3 LSTM Confusion Matrix

```
Joy -
[[ 5371  2127]
 [ 5155 63308]]

Anger -
[[ 5498  2000]
 [ 5686 62777]]

Surprise -
[[ 5394  2104]
 [ 4873 63590]]

Fear -
[[ 5389  2109]
 [ 5459 63004]]

Contempt -
[[ 5264  2234]
 [ 4665 63798]]

Disgust -
[[ 5391  2107]
 [ 5401 63062]]
```

```
Sadness -
[[ 5582  1916]
 [ 5970 62493]]

Confusion -
[[ 5443  2055]
 [ 5039 63424]]

Frustration -
[[ 5274  2224]
 [ 4840 63623]]

Neutral -
[[ 5551  1947]
 [ 5763 62700]]

Positive -
[[ 5321  2177]
 [ 5056 63407]]

Negative -
[[ 5348  2150]
 [ 5156 63307]]
```

5 Toolbox

I have used Python deep learning libraries for producing and training the neural networks I have used in this project. I have used Tensorflow and Keras from the deep learning libraries. For boosted trees and linear models I have used the xgboost and sklearn packages respectively.

6 Code and Results

https://github.com/bmukheja/Machine-Learning-Emotion-Data

References

[1] https://www.tensorflow.org/

[2] https://github.com/fchollet/keras

[3] Godbole S., Sarawagi S. (2004) Discriminative Methods for Multi-labeled Classification. In: Dai H., Srikant R., Zhang C. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2004. Lecture Notes in Computer Science, vol 3056. Springer, Berlin, Heidelberg