Thesis Report 4 : 09 March - 16 March

Goals

- Improve LMA-Emotion regression model(s) ✅
- Integrate Emotion identification into project ✅
- Research models to convert Emotion coordinates to LMA features ✅

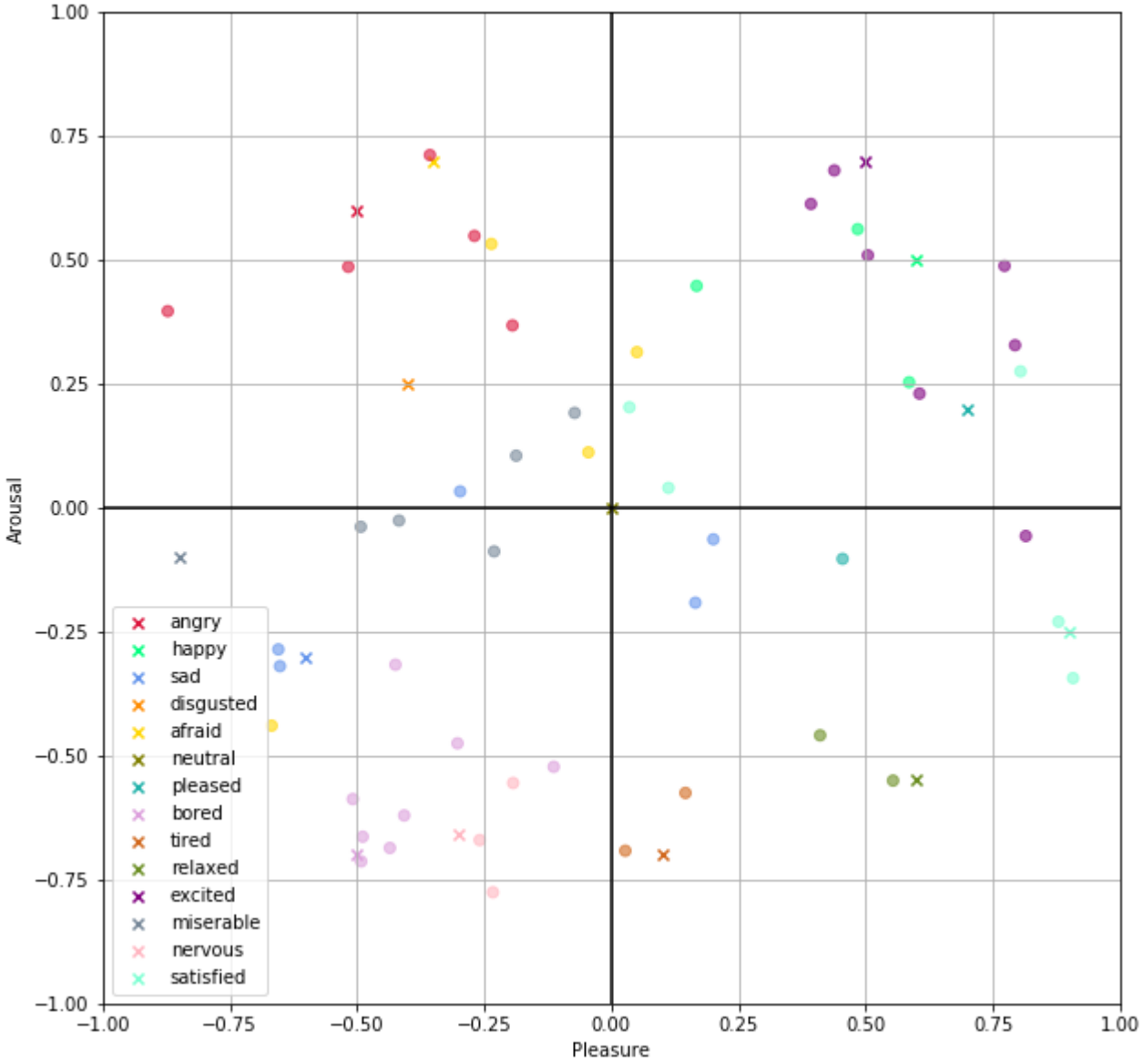
Last Week Leftovers:

None

Done

- Continued experimenting with different neural network configurations to use an MLP to create regression
    - Results wielded weren't too great, so didn't spend too much time on this and moved on to XGBoostRegression
- Altered PA coordinates to make them more hopefully more accurate
    - "angry": (-0.5, 0.6),
    - "happy": (0.6, 0.5),
    - "sad": (-0.6, -0.3),
    - "disgusted": (-0.4, 0.25) ,
    - "afraid": (-0.35, 0.7),
    - "neutral": (0.0, 0.0),
    - "pleased": (0.7, 0.2),
    - "bored": (-0.5, -0.7),
    - "tired": (0.1, -0.7),
    - "relaxed": (0.6, -0.55),
    - "excited": (0.5, 0.7)
    - "miserable": (-0.85, -0.1),
    - "nervous": (-0.3, -0.66),
    - "satisfied": (0.9, -0.25),

- Also added the Dominance coordinate. However, there is not much information online (that I could find) that indicated reference dominance values for these emotions, so most of them came out of intuition. **If possible, I would like the coordinators to help me figure out the best dominance values for each of these emotions!!**. It should be noted that, after having the dominance coordinates well defined, the regressor that's being used for the pleasure and arousal coordinates can be easily fit to also compute the dominance coordinate.
    - "angry": (-0.5, 0.6, 0.9),
    - "happy": (0.6, 0.5, 0.2),
    - "sad": (-0.6, -0.3, -0.3),
    - "disgusted": (-0.4, 0.25, 0.0) ,
    - "afraid": (-0.35, 0.7, -0.8),
    - "pleased": (0.7, 0.2, 0.2),
    - "bored": (-0.5, -0.7, -0.25),
    - "tired": (0.1, -0.7, -0.2),
    - "relaxed": (0.6, -0.55, 0.1),
    - "excited": (0.5, 0.7, 0.4),
    - "miserable": (-0.85, -0.1, -0.8),
    - "nervous": (-0.3, -0.66, -0.7),
    - "satisfied": (0.9, -0.25, 0.65),
- **Kept training the XGBoostRegression model**. Managed to get really low MAE (~0.10) and was really happy about it. But then I realized I was accidentally computing the training data's MAE rather than the test data. Actual MAE values were around ~0.25 (with MSE's around 0.11) for either coordinates. **Our model was overfitting by a lot**
- Connected Jupyter Notebook to remote server and started training models there (using one of the 2 GPUs on the server)
- **Performed a GridSearch** to try to find the best parameter, using a 10-fold Cross Validation. Results yielded were pretty poor.



- Due to poor results, started reanalyzing both the Emotional Control of Unstructured Dance Movements paper and our own mocap data.
    - Remembered that despite +- good results on the paper, they did admit that *"As we can see, the emotion coordinates do not really align with each other or the original position from the RCM model marked by the blue dots. This implies the complexity of emotions. However, all the predicted emotion coordinates do fall into the correct quadrant of the RCM diagram."*.
    - Furthermore, I came to that conclusion that, despite labeling all "sad" mocap with the same emotional coordinates, all "afraid" mocap with the same emotional coordinates, and so on and so forth, each mocap performance displays a different level of "sadness" and "afraidness" and so on. As such, it should be expected that the values do diverge a bit.
    - Moreover, even if the performance of the mocap was "sadness", we extract LMA features over a window of 30 frames, and its normal that there are moments within the animation itself that are more or less sad. However, we label them all with the same sadness value,

which may hinder the regression.
    - Also notice that one of our mocap datasets - the largest one (Kin) - had a lot of artifacts when converted to the Deepmimic format.

- Taking all of this into account (overfitting plus other conclusions), decided to remove all mocap from the Kinematic dataset from our data (we now have mocap from the Dance database and some Walk animations). We went from ~50000 entries to about ~10000. Before doing this I tried adding LMA features, removing them and tried out a bunch of data preparation to try to salvage whatever I could, but to no avail.

- Retrained XGBoost regression with simple parameters to the new reduced dataset (after performing data preparation on it - normalization/removing highly correlated features/balancing) and wielded positive results

- **Left remote server doing another extensive GridSearch. Still waiting on the results**

- **Started integrating emotion classification to our actual project**. Right now, by specifying an argument when starting the spacetimebounds vis_mocap/run_model scripts, we record all LMA features (again over a window of 60frames). When the animation is over (or it loops) we pass them to our EmotionClassifier, which then takes them and computes the coordinates of each of these recorded feature sets. It then computes the average coordinates and prints them out on the console.

    - This is a good first start, but there are issues with this algorithm. Namely:
        - The fact that we have to wait for the animation to complete/loop - instead it would be better to get X number of feature sets and after that number is achieved run the emotion classification algorithm (e.g every time we extract 10 sets of LMA features, we run)
        - Because we're only classifying at the end of the animation, the algorithm does take a while to normalize all LMA features and to classify them one by one. If we normalized and classified the features as they came in, rather than as a batch at the end, this may make classification much quicker (i.e rather than taking 10 seconds at the end to classify everything, we take 0.1seconds each frame to do it)

- Did some research on possible network architectures to solve the problem of going from 2 (or 3) emotional coordinates to a larger set of LMA features (i.e emotion-motion mapping). Found **GANs** - Generative Adversarial Networks.

    - *"Two neural networks contest with each other in a game (in the form of a zero sum game, where one agent's gain is another agent's loss). Given a training set, this technique learns to generate new data with the same statistics as the training set. For example, a GAN trained on photographs can generate new photographs that look at least superficially authentic to human observers, having many realistic characteristics."*

- Maybe try to use an autoencoder instead of a GANs

Left Undone

There was a Grid search training left going that was not completed at the time of writing. Need to check the params it found and if that performs better than what we already have

Problems

Notes

Thoughts

This week was full of peaks and valleys. I was super happy when I got a model with 0.11 MAE on both pleasure and arousal. Then I was devastated when I realized my stupidity and error of having computed the errors using my training set rather than test. Then I panicked, started trying a bunch of stuff, doing research and all that. Taking everything into account - complexity of the problem, our datasets and other efforts that have been made at trying to map LMA to emotion coordinates - I don't think our MAEs will ever be lower than 0.1. Regardless I am trying my best to get them to round the 0.15 house and hope that, by doing an average of predictions when identifying an emotion from a full motion, the predicted values will round out to be more or less the real emotion. Nevertheless, I believe what we have now is "good enough" and will still move on to working on to the Emotion-Motion mapping during next week. If I have time next month I will revisit the Motion-Emotion mapping models and try to make them better.

Work Hours

- Worked everyday but Sunday from 1pm to 7pm