Thesis Report 7 : 30 March - 13 April

Goals

- Redid LMA feature extraction algorithm ✅
- Reextract LMA features ✅
- Improve LMA-PAD regression ✅
- Improve Emotion Classification Algorithm integration ✅
- Work on PAD-LMA regression ✅
- Autoencoder for LMA feature generation ✅

Last Week Leftovers:

None

Done

- Added more LMA features (related to movement) and slightly altered others, especially in regards to **movement/speeds**

```
{
    frame_counter: frame at which LMA features were computed,

    label: PAD Emotional Coordinates (3D)

    lma_features: [
        (CHANGED) max hand_distance (1D),
        average l_hand_hip_distance (1D),
        average r_hand_hip_distance (1D),
        (CHANGED) max stride length (feet distance) (1D),
        average l_hand_chest_distance (1D),
        average r_hand_chest_distance (1D),
        average l_elbow_hip_distance (1D),
        average r_elbow_hip_distance (1D),
        average chest_pelvis_distance (1D),
        average neck_chest_distance (1D),

        average neck_rotation (4D),
        (CHANGED) average total_body_volume (1D),

        (NEW) average area of triangle between hands and neck (1D),
        (NEW) average area of triangle between feet and hips (1D),
```

```
        # Speed: Position Change between first and last frame / interval between lma
        (NEW) average l_hand speed  (1D),
        (NEW) average r_hand speed (1D),
        (NEW) average l_foot speed (1D),
        (NEW) average r_foot speed (1D),
        (NEW) average neck speed (1D),

        # Acceleration: Linear Velocity Change between last frame of previous record
        (NEW) l_hand acceleration magnitude (1D)
        (NEW) r_hand acceleration magnitude (1D)
        (NEW) l_foot acceleration magnitude (1D)
        (NEW) r_foot acceleration magnitude (1D)
        (NEW) neck acceleration magnitude (1D)

        # Movement Jerk: Acceleration Change between acceleration in last frame of p
        (NEW) l_hand movement jerk (3D)
        (NEW) r_hand movement jerk (3D)
        (NEW) l_foot movement jerk (3D)
        (NEW) r_foot movement jerk (3D)
        (NEW) neck movement jerk (3D)
    ]
}
```

- Changed the **LMA the extraction algorithm**:
    - Before now we extracted LMA features every 30 Frames, regardless of the FPS of the animation. This could be problematic because of **animations with wildly different FPS!**.
    - For example, imagine we have 2 animations, one that's 60 FPS and one that's 30 FPS. If we extract LMA features over a 30frame window, for the first animation we're getting data over 0.5 seconds, whilst for the second we're getting data over 1 seconds.
    - **This was solved by extracting features over X seconds rather than X frames** (e.g extracting each 1 second of animation for animation 1 would mean every 60 Frames and for animation 2 each 30 Frames). Note that we're able to infer the animation's FPS by getting each frame's duration and doing 1/frame_duration.
- Re-extracted LMA features to retrain models
- Trained XGBoostRegression for LMA-PAD mapping using new features (had to re-do data profiling and data preparation). Tried out LMA features extracted over 0.5 seconds and 1 second.
    - Noted that we had 2 emotions (out of our set of 13) - Nervous & Bored - that were very underrepresented.
    - To reduce this class imbalance we used to use SMOTE to generate new samples. But since the imbalance was too large, even after performing some under sampling on the

more represented emotions, we were creating too many artificial samples which were hindering our learning process. So we just removed those emotions outright.

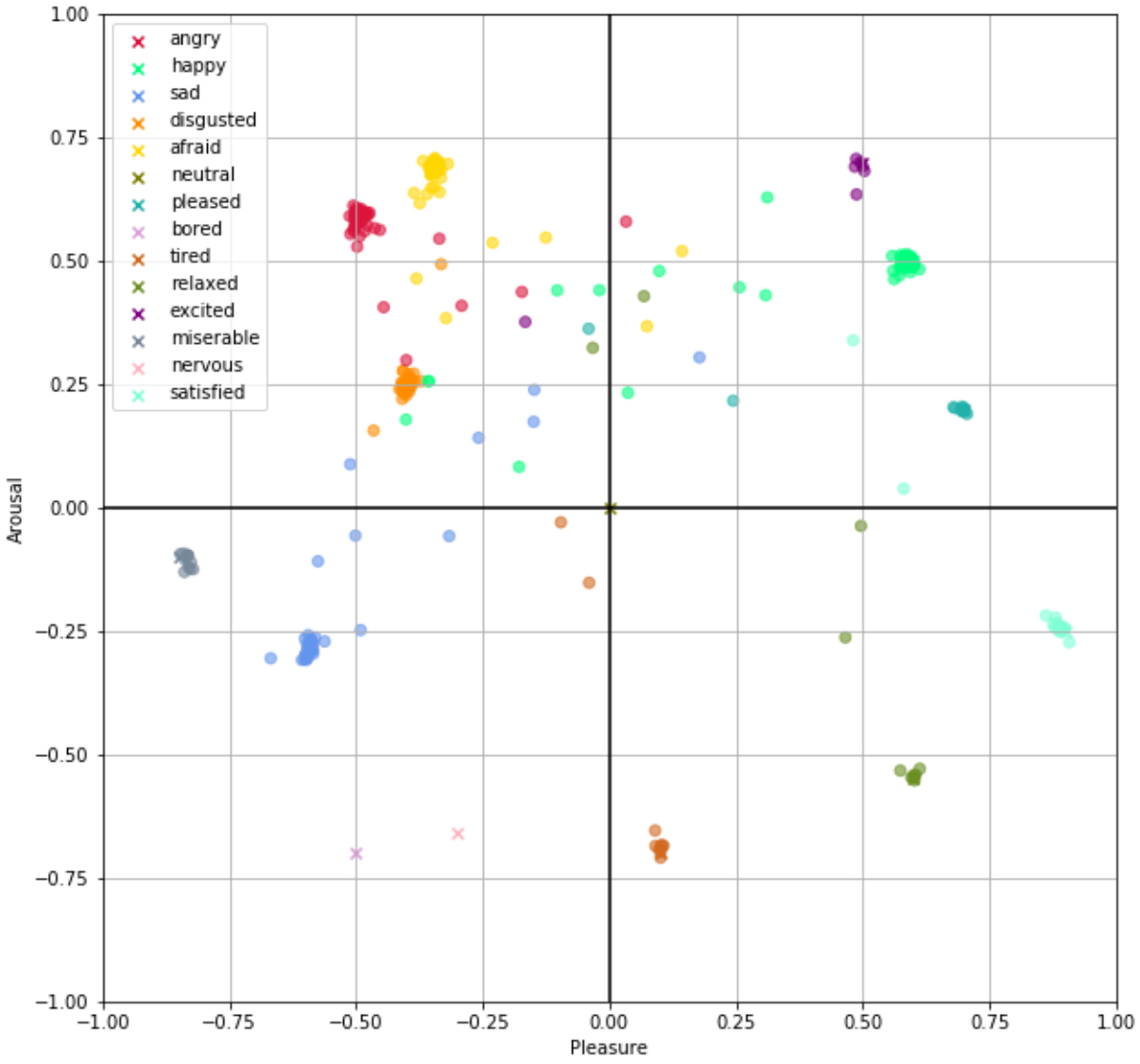
- **Managed to improve MAE and MSE DRASTICALLY of our LMA-PAD regression using the 0.5sec interval LMA features**and RandomGridSearch with 5-fold cross validation to tune our XGBoostRegression hyperparameters

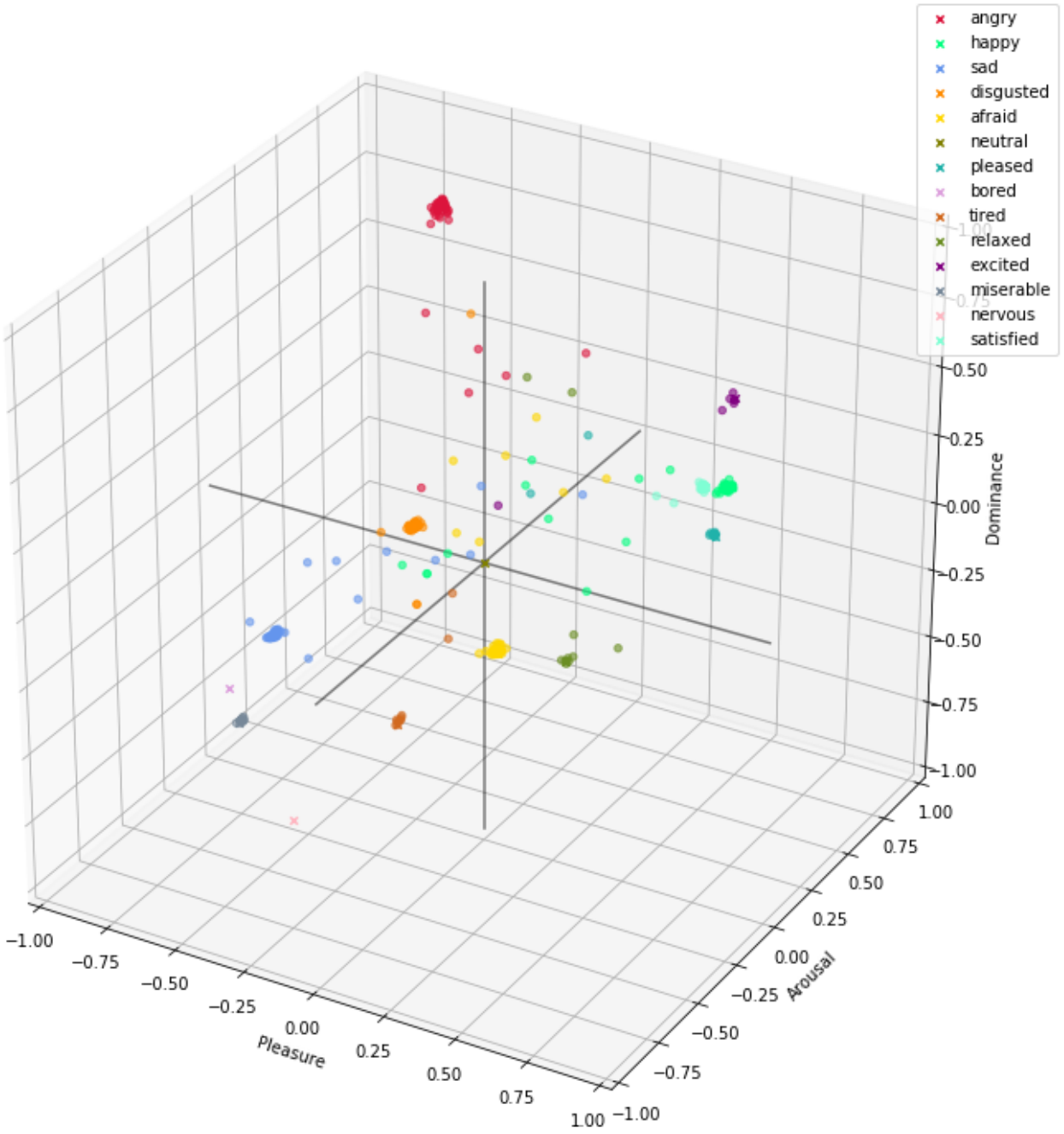
```
Pleasure
MSE: 0.03
MAE: 0.07

Arousal
MSE: 0.02
MAE: 0.06

Arousal
MSE: 0.04
MAE: 0.07
```

  - **Reduced all errors to under 0.7!**





- Improved Emotion Classification algorithm
    - Made it so instead of averaging each set of 10 LMA features, we now instead store all of them (averaging was making it so we diluted the peaks of emotions too much)
- Tried creating an LSTM similar to the one used by https://arxiv.org/pdf/1906.11884.pdf but after further reading found out that **they don't actually use an LSTM for emotional classification**
    - What the authors did was take videos of walking gaits (rather than actual Mocap data), and used an LSTM in order to extract joint positions and LMA features from the videos. Classification/regression was then done using a Random Tree Forest
    - Whilst LSTM's may still be worth exploring, I came up with another idea I wanted to try out

before (see the next bullet point)

- Since our Emotional Classification algorithm is already performing pretty well, I think I've settled on disregarding using something other than the XGBoostRegressor for LMA-PAD mapping

- Using the new dataset, re-implemented the PAD-LMA regression using XGBoostRegression
  - Using the new LMA features, and toying a bit with hyperparameters we were able to create a set of Regressors that manage to generate each individual LMA feature (from our set of 27) given the PAD coordinates
  - MAEs and MSEs of each generated feature are *decent*. Some manage to achieve <0.15 MAE. However, others do much worse.
  - This may be improved by continuing to play with the hyperparameters, or by simply **not using the generated LMA features with higher errors (which mainly constitute of features pertaining to Acceleration Magnitude) for our motion synthesis**

- Created a Variational AutoEncoder to generate new LMA features.
  - Our previously stated problem with Variational AutoEncoders was that we couldn't feasibly have a latent space that corresponded to the PAD coordinates, since compressing from ~30 features down to 3 would make it so the decoder couldn't reliably go back from latent space to real space
  - Furthermore, the autoencoder is the one that finds the optimal encoding, meaning we can't really force it to use the PAD coordinates as the compression
  - So my idea was: **train an autoencoder to be able to generate LMA features**. Then, **use XGBoostRegression to be able to map PAD coordinates into the Latent Space, so that the decoder can then take that latent space and use it to generate the LMA features**. This way we would still be taking advantage of the autoencoder's generative capabilities, and circumventing the downsides that were preventing us from using it for our specific problem.
  - Generated features have been decent, but they're still slightly worse than the ones generated by our previously stated regression. I'm gonna keep adjusting the hyperparameters (i.e network architectures) of our autoencoder to try to improve it during this next week
  - Similarly, I still have to try to create a regressor to go from the PAD coordinates into latent space (which currently has a size of 5) , which I will be trying to do during this week
  - *Note: Out of curiosity I also tried training a GANs to generate LMA features. It performed +- better than the autoencoder on the generation task, but since we can't actually get a mapping between the real sample and its latent space correspondent (something we can do on the autoencoder due to the Encoder portion of the network), we can't really map PAD coordinates into latent space*

Left Undone

Problems

Notes

Thoughts

So this week I had to make the harsh decision to go back and redo the entire LMA feature extraction algorithm. I was debating whether this was worth it or not, but in the spirit of investigation I thought I should go for it, even if it took me a couple of days to rewrite the way features were extracted, which features we extracted, and then actually leaving the remote server extracting the features. But I'm (obviously) glad I did so.

I was honestly not expecting that the emotion identification errors could be further minimized, but lo and behold, using this new and improved LMA extraction, and by doing some hyperparameter tuning/data preparation, we managed to get errors below 0.07! That's honestly amazing, taking into account that our values vary between -1.00 and 1.00, that small of an error is pretty good. With this, and after also having imrpoved the emotion classification integration with the motion learning, I think I'm effectively 100% done with the **emotion detection** half of this project. Obviously the error isn't 0, and there's always room for improvement/experimentation, but I'm going to shift my entire focus into the remaining aspects of the project, with the prospect of being done with all coding/implementation by the end of May (so that June/July are free for writing the thesis document and performing all user testing necessary).

So moving onto LMA feature generation and Motion Synthesis. We got some decent models using our regression, in the same vain as the Emotion Control of Unstrucutred Dance Movement paper. However, in the meantime I had an idea that might make it so we could actually use Variational AutoEncoders for the generation, so I started training some autoencoders. This is also what I'm expecting to do throughout this next week.

By the end of next week I want to have my LMA generation from PAD coordinates model(s) effectively finalized, so that I can move on to actual algorithmic Motion Synthesis (which I'm expecting to be working on from the 20th of April, all the way until the 19th of May ~3/4 weeks). I've got a few ideas floating around my head as to how Motion Synthesis is actually going to be done, but for now I first wanna get a decent model to actually give me the LMA features that will serve as the baseline for the synthesis.

Work Hours

- Worked during my off-week - Managed to squeeze in some hours of work during my self-proclaimed off-week which I used to redo the LMA extraction algorithm

- Worked each day from the 4th to 9th from 1pm until 4pm
- Worked 11th and 12th from 1pm until 8pm