AMATH 582: HOMEWORK 2

HUNTER LYBBERT

Applied Mathematics Department, University of Washington, Seattle, WA
hlybbert@uw.edu

ABSTRACT. This is my Abstract **TODO**

1. Introduction and Overview

This is my Introduction **TODO** In order to understand the data and attempt to classify the movements of the robot we aimed to complete the following 5 tasks:

(1) Through **averaging of the Fourier transform** determine the dominant frequency (center frequency) generated by the submarine. Verify your results through visualization.

(2) Design and implement a **Filter** to extract this center frequency in order to denoise the data and determine a more robust path of the submarine. Visualize the denoised measurement the 3D path of the submarine and inspect the validity and effectiveness of the denoising.

(3) Determine and **plot the** x-y coordinates of the submarine path during the 24 hour period. This information can be used to deploy a sub-tracking aircraft to keep an eye on your submarine in the future.

In the endeavor to reduce the dimensionality of the robot movement data, classify the movements, and complete the requisite tasks, we made extensive use of several important Python packages. Namely, Matplotlib was used to create all plots and animations [2]. Additionally, Scikit-learn was the primary source of using the PCA algorithm and other classification methods [3]. Finally, NumPy was once again a crucial tool [1]. Moreover there are important theoretical underpinnings behind the algorithm we implemented which will be cited and expounded upon in the following section.

2. Theoretical Background

Technical background duh duh duh ... **TODO**

$$X = U\Sigma V^T \tag{1}$$

Let's get into the actual implementation now.

3. Algorithm Implementation and Development

We will now describe in words and pseudocode the implementation of these methods as we used them in this application to reduce the dimensions of our data and try to preserve as much information as possible. First, we discuss the PCA algorithm in 1.

Date: February 4, 2025.

Algorithm 1 Determine the Dominant Frequency

$S = \text{np.load(data)}$ ▷ Input subdata, after reshaping to (64, 64, 64, 49)
$\hat{S} = \text{fftn(S)}$
$\hat{S}^{\dagger} = \text{fftshift}(\hat{S})$ ▷ Transformed and shifted subdata (64,64,64,49)
$\hat{S}^{\dagger}_{\text{avg}} = \text{avg}(\hat{S}^{\dagger})$ ▷ Average computed across time at each point (64,64,64)
$x, y, z = \text{argmax}\left(\text{abs}\left(\hat{S}^{\dagger}_{\text{avg}}\right)\right)$

The result of the final step of Algorithm 1 are **TODO: fill in deets**. What we really want is to know what the frequency value should be to center our gaussian filter from equation (1) to use in Algorithm 2

Algorithm 2 Apply Gaussian Filter in Frequency Space

$G = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(k_x - k_{x_0}\right) + \left(k_y - k_{y_0}\right) + \left(k_z - k_{z_0}\right)\right)$ ▷ Gaussian Filter using (1)
$\hat{S} = \text{fftn(S)}$
$\hat{S}^{\dagger} = \text{fftshift}(\hat{S})$
$\hat{F}^{\dagger} = \hat{S}^{\dagger} G$ ▷ Filtered subdata in frequency space still
$\hat{F} = \text{ifftshift}(\hat{F}^{\dagger})$
$F = \text{ifftn}(\hat{F})$ ▷ Filtered subdata in signal space now

TODO:. These results will be described further in following section.

4. Computational Results

We have the following to talk about:

- Talk about getting the right shape for training
- That means we had to treat each frame of the robot movements as a single sample
- Talk about the classification issues
- why we used the Support vector machine to try and classify more accurately

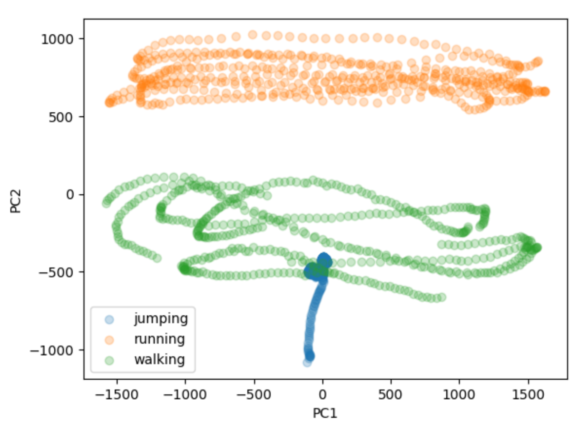

FIGURE 1. Visualizing each of the 3 slices including the location in our 3 dimensional average frequency object.

As seen in Figures 1 and 2, then Figure 3.

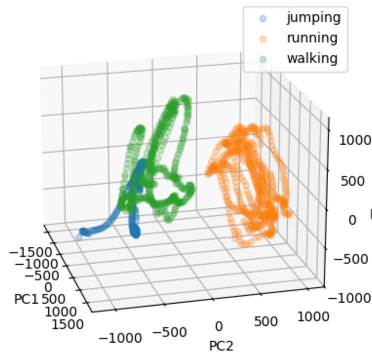

Figure 2. Visualizing each of the 3 slices including the location in our 3 dimensional average frequency object. We also visualize a slice which does not intersect the max frequency in order to convey how drastic the location of the max frequency is. In order to show this comparison things have been rescaled here.

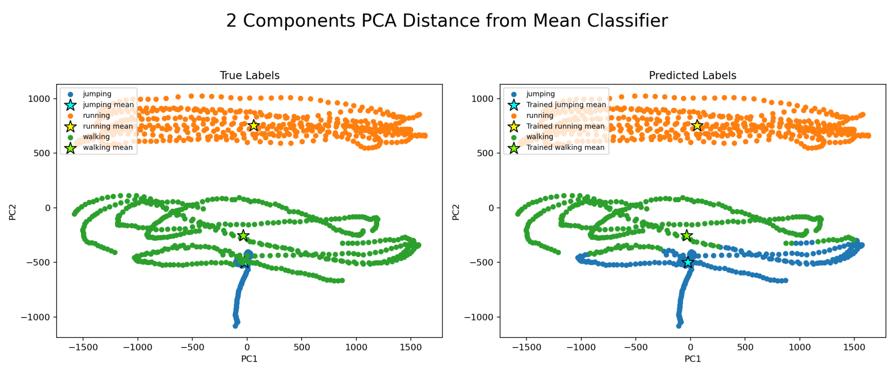

Figure 3. The resulting path in 3 dimensions that we determined after applying Algorithms **??** and **??**. In this iteration of the filter we used $\sigma = 1.3$.

5. Summary and Conclusions

This is my summary and conc.

Acknowledgements

The author is thankful to Jaxon Tuggle for useful discussions about the process to find the correct dimensions to use when interpreting the training data samples we were given. We would also like to thank Professor Eli Shlizerman for carefully instructing us in class. Finally, it is necessary to thank the following students Nate Ward, Sophie Kamien, whose questions helped clarify understanding of the algorithm we implemented by giving chances to explain ideas and debug code together.

References

[1] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020.

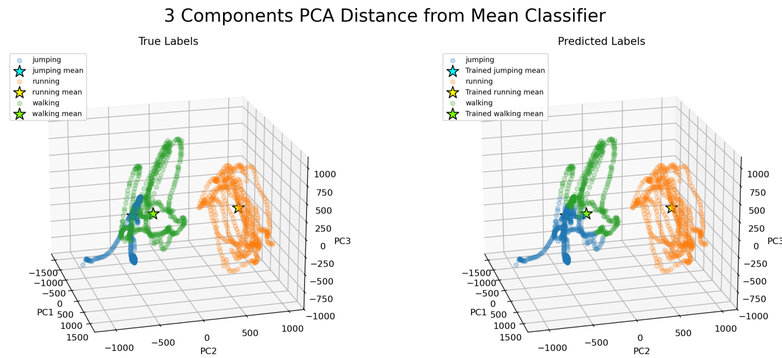

FIGURE 4. This is just the 2 dimensional projection of the path given in the above 3d plot. The same value of σ was used in the filter.

[2] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.