AMATH 482 Homework 2

Trevor Ruggeri

Due: February 19, 2025

This report presents the development and implementation of a classification algorithm for humanoid robot OptimuS-VD's movement recognition. OptimuS-VD records joint movements as Euler angles, which are transformed into xyz coordinates. PCA is applied to reduce the dimensionality of movement data, enabling visualization and classification of walking, jumping, and running. The following report evaluates the effectiveness of different PCA truncations in retaining energy, establishes ground truth labels, and trains a classifier based on centroids in PCA space. Finally, test samples are classified, and accuracy is analyzed.

1 Introduction/Overview

The OptimuS-VD humanoid robot is designed to perform fundamental movements: walking, jumping, and running. The robot records the xyz coordinates of 38 joints at 60Hz over 1.4 seconds, forming a 114x100 matrix for each movement sample. This study applies Principal Component Analysis (PCA) to project the high-dimensional movement data into a lower-dimensional space, facilitating visualization and classification. The project aims to:

- Reduce dimensionality while preserving essential movement information.
- Visualize the movements in a lower-dimensional PCA space.
- Train a classifier using centroid-based distance minimization.
- Evaluate classification performance on test samples.

2 Theoretical Background

2.1 Principal Component Analysis (PCA):

PCA is a statistical technique that transforms high-dimensional data into a lower-dimensional space while preserving variance. It decomposes the dataset into spatial modes (eigenvectors) and time-dependent coefficients (eigenvalues). Using Singular-Value-Decomposition, we can write our data matrix X as

$$X = U\Sigma V^T,$$

where U contains the left singular vectors (spatial modes), Σ is the diagonal matrix of singular values, and V^T contains the right singular vectors (time-dependent coefficients).

2.2 Classification in PCA Space:

By computing centroids for each movement class in the reduced PCA space, a nearest-centroid classification approach is applied. The trained classifier assigns movement labels based on the minimal distance between projected test samples and the centroids. Given a new sample x, it is assigned to a category c by

$$\underset{c}{\operatorname{argmin}}\ ||x - \mu_c||,$$

where μ_c is the centroid of category c in k-mode PCA space.

3 Algorithm Implementation and Development

- numpy: Numerical computations
- sklearn.decomposition.PCA: Principal Component Analysis
- sklearn.metrics.accuracy_score: Classification evaluation
- plotly.express: Visualization

3.1 PCA Analysis on Training Data:

- Combine all training samples into matrix.
- Apply PCA and determine the number of modes needed to retain 70%, 80%, 90%, and 95% energy.
- Plot cumulative energy to justify truncation choices.

3.2 Visualization of Movement in PCA Space:

- Truncate PCA to 2 and 3 modes.
- Plot movement trajectories in 2D and 3D PCA space using different colors for each movement.

3.3 Establishing Ground Truth Labels and Computing Centroids:

- Assign integer labels: 0 (walking), 1 (jumping), 2 (running).
- Compute class centroids in k-modes PCA space.

3.4 Training and Classification:

- Compute distances from each sample to centroids.
- Assign trained labels based on the closest centroid.
- Evaluate classification accuracy for various values.

3.5 Testing on New Samples:

- Project test samples into PCA space.
- Assign movement labels based on the nearest centroid.
- Compare test accuracy with training accuracy.

3.6 Building Logistic Regression Classifier and Comparing Performance:

- Project training samples into PCA space.
- Use sklearn.model_selection.train_test_split() to further split training data into training and testing data.
- Build a logistic regression model and compare accuracies to determine optimal k-value
- Calculate accuracy of logistic regression classifier in optimal k-mode PCA space on test set and compare with centroid classifier

4 Computational Results

4.1 Approximating Training Data up to 70%, 80%, 90%, 95% Accuracy

When performing PCA, the dimension reduction there is (the more PCA spatial modes are used), the better the original data is represented. We transformed the data to k-modes PCA space using PCA.fit_transform(), taking between $k = 1$ and $k = 114$ spatial modes. Using PCA.inverse_transform(), we transformed the PCA-data back to our real-world space, and measured the accuracy of the recovered data using the Frobenius norm. Graphed below are the Frobenius norms for different k-values. In Figure 1 above, we see that the

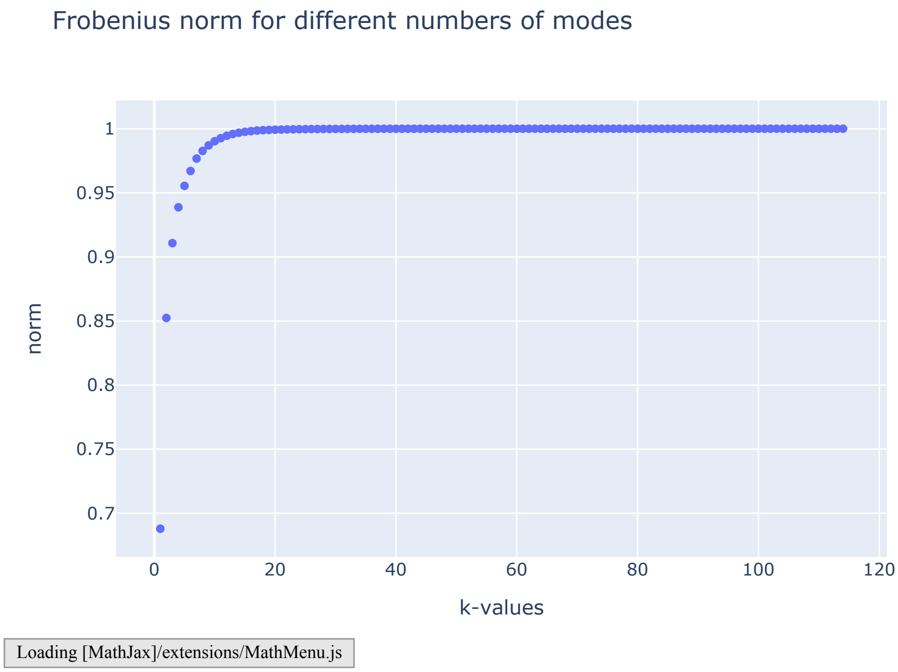


Figure 1: Frobenius Norm for Different Numbers of Modes

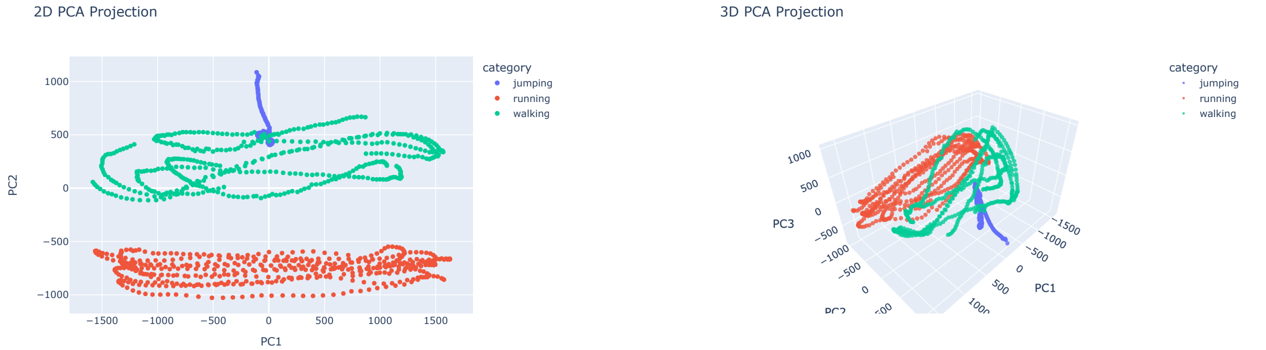
Frobenius norm starts below 70% for 1 spatial mode, and quickly rises to 100% around 18 spatial modes. Specifically, the Frobenius norms for 1-5 spatial modes are listed in the table below. We can see that 2 spatial modes are needed for 80% energy, 3 modes are needed for 90% energy, and 5 modes are needed for 95% energy.

k-value (spatial modes)	Frobenius norm (energy)
1	0.688012
2	0.852402
3	0.910751
4	0.938725
5	0.955401

Table 1: Frobenius Norms for 1-5 spatial modes

4.2 Visualizing Data in 2D and 3D PCA space

Focusing on 2- and 3-mode truncations of the PCA data, we can visualize the three different movements graphically. Figure 2 shows the 2D and 3D truncations of the PCA data, with each point color-coded to represent its associated movement. In Figure 2a and Figure 2b, we can see that in both 2D and 3D PCA



(a) 2D PCA Truncation

(b) 3D PCA Truncation

Figure 2: Truncations of the PCA data showing each movement

space, the jumping, running, and walking data points form distinct clusters. These data clusters in PCA space are used later to classify new data points, as we can associate an unknown movement point with the cluster closest to it.

4.3 Finding centroid of each movement in k-modes PCA space

By establishing a vector of ground truth labels for each of the movements in PCA space, we can calculate the centroid of each class of movements by computing the mean of the points in k-mode PCA space. Table 2 shows the centroid coordinates in 5-mode PCA space of each of the three movements

Movement	PC1	PC2	PC3	PC4	PC5
Jumping	-23.88986635	499.36826149	-72.5000755	28.58077286	1.89232891
Running	60.77197779	-752.7210869	-103.41194553	1.71007194	-43.51746561
Walking	-36.88211143	253.35282541	175.91202104	-30.2908448	41.62513671

Table 2: PC1, . . . , PC5 coordinates for centroid of each movement

4.4 Training Classifier using centroid method

Using the centroids calculated above, we can train a classifier that takes new data in k-mode PCA space and assigns it a class based on the closest centroid. Using accuracy_score() from sklearn.metrics, we can calculate the accuracy of this classifier on the training data. Figure 3 below shows the accuracy of the classifier on the training data for different k-values.
Specifically, Table 3 below shows these accuracy scores for different k-values.
Since our goal is to maximize accuracy using as few modes as possible, it appears using $k = 2$ modes for our PCA is the optimal strategy.

4.5 Testing classifier on testing data

Using the same process as above, we can calculate the accuracy of our classifier on the testing data. Shown in Table 4 are the accuracies of the classifier on the testing data for different numbers of modes.

4.6 Training Logistic Regression Classifier in k-mode PCA space

For each k-value between 1 and 114, we split the training data using sklearn.model_selection.train_test_split() and trained a logistic regression classifier from the sklearn.linear_model package. Figure 4 shows the accuracy

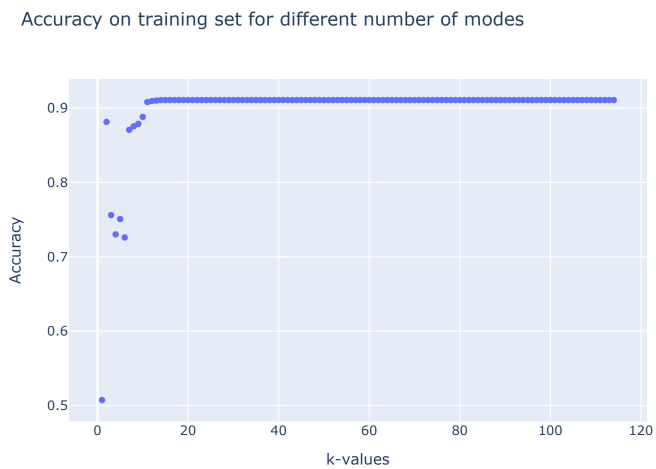


Figure 3: Accuracy of Classifier on Training Data for different k-values

k-value	Accuracy
1	0.507333
2	0.881333
3	0.756
4	0.73
5	0.750667
6	0.726
7	0.870667
8	0.875333
9	0.878667
10	0.888

Table 3: Accuracy score of classifier on training for different number of modes

of the classifier on the k-mode PCA data for k-values between 1 and 20, as measured by its accuracy on the testing portion of the split training data.

From Figure 4, we can see that using 2 spatial modes has a high accuracy on the training data while minimizing the number of dimensions used. However, using this trained model on the testing data in 2-mode PCA space yielded an accuracy of 0.23333.

5 Summary and Conclusions

This study successfully applied PCA to project high-dimensional movement data of OptimuS-VD into a lower-dimensional space, enabling visualization and classification. The key findings include:

- The number of PCA modes required to retain significant movement information were low, as 5 modes still captured over 95% of the movement information.
- Effectiveness of centroid-based classification in PCA space proved successful, as in 2-mode PCA space the centroid-based classification had over a 90% accuracy.
- The logistic regression classifier in k-mode PCA space proved less effective than the centroid-based classifer, as a far lower accuracy on the test data was reported.

  Future work could involve more sophisticated classification techniques, such as neural networks, to improve recognition accuracy.

k-value	Accuracy
1	0.506667
2	0.933333
3	1.0
4	1.0
5	1.0

Table 4: Accuracy score of classifier on test data for different number of modes

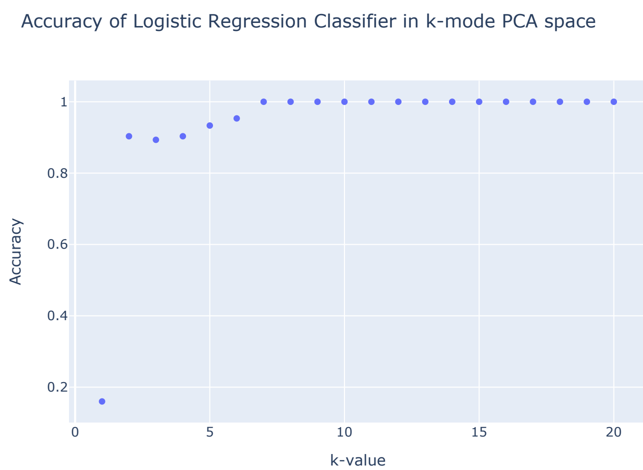


Figure 4: Accuracy of logistic regression classifier on split training data for different k-values

5.1 Acknowledgements

I would like to thank Rohin Gilman for providing me with valuable insights and tips when constructing my code for this project, as well as helping me understand some of the theoretical concepts behind what we were doing.

5.2 References

Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2.

Kutz, J.N. (2013) Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data. Oxford: Oxford University Press.