AMATH 582: HOME WORK 2

SUKHJIT KAUR

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

Abstract. Sensor data from 38 joints, recorded at 60Hz as 3D coordinates, capture walking, jumping, and running movements over 1.4 seconds, for a humanoid robot named OptimuS-VD. Using Principal Component Analysis (PCA), the high-dimensional data is projected into a lower-dimensional space, enabling visualization and the development of a centroid-based classifier. The model is tested on unseen movement samples for validation.

1. Introduction and Overview

Humanoid robots rely on precise motion control and recognition to perform complex tasks. Understanding and classifying their movements is essential for improving autonomy, adaptability, and human-machine interaction. OptimuS-VD is equipped with sensors that record the positions of 38 joints at a frequency of 60Hz, capturing movement as Euler angles, which is transformed into 3-dimensional coordinates. The collected data consists of three distinct movement types—walking, jumping, and running— each recorded over 1.4 seconds with 100 timesteps. The dataset is structured as a 114×100 matrix, where each row represents the x-, y-, and z-coordinates of all joints, and each column corresponds to a timestep.

Due to the high dimensionality of the motion data, analyzing and visualizing it directly is challenging. Principal Component Analysis (PCA) is employed for dimensionality reduction. PCA is a statistical technique that transforms the original high-dimensional data into a lower-dimensional space while preserving the variance of the data as much as possible. It identifies the principal components—orthogonal directions in which the data exhibits the most variation—allowing a compact and informative representation of patterns. By projecting the motion data onto a lower-dimensional subspace, PCA allows visualization and reveals inherent structure and differences between movement types.

This reduced dimensionality serves as the foundation to build a classification algorithm that can recognize movements in real-time. By training the model on projected motion data, it becomes possible to classify new movement samples based on their principal component representation.

2. Theoretical Background

At the core of PCA is Singular Value Decomposition (SVD), a matrix factorization method used to compute the principal components. Given an m × n data matrix X, SVD decomposes X as, [1].

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

where U is a matrix whose columns are the left-singular vectors and represents the basis vectors in the high-dimensional space, Σ is the diagonal matrix containing singular values that represent

Date: February 11, 2025.

the variance of each component, and V^T is a matrix whose rows are the right singular vectors, representing the directions of the principal components. The variance is a measure of how much a set of values deviates from their central value (mean). It is calculated as follows:
[1]

$$\sigma^2 = \frac{1}{N}\Sigma_{i=1}^{N}(x_i - \mu)^2 \tag{2}$$

For a multidimensional dataset, we can use the covariance matrix, which summarizes the relationships between all the variables, and is computed as:

$$C_x = \frac{1}{N-1}XX^T \tag{3}$$

Applying SVD to the covariance matrix yields an approximation as follows (1) on the matrix, X, where U represents the PC-modes of C_x:

$$C_x \approx \frac{1}{N-1}U\Sigma^2U^T \tag{4}$$

PCA uses SVD by choosing the top k singular values and their corresponding vectors, effectively reducing the data's dimensionality while preserving its most significant variance. This allows the data to be projected onto a lower-dimensional subspace, simplifying visualization, and allowing for classification of new data based off the meaningful correlation found.

3. Algorithm Implementation and Development

All recorded movement samples in the data were loaded into a matrix X_{train} of shape (15, 114, 100), where each sample contained the (x, y, z) coordinates of 38 joints over 100 time-steps. The data was pre-processed by reshaping into a 2D matrix of shape (1500, 114), where each time-step is an independent observation. The ground truth labels were assigned as 0 (walking), 1 (jumping), and 2 (running), and repeated for each time step.

Scikit-learn's PCA implements SVD internally for dimension reduction [2]. scikit-learn's PCA was applied to reduce the dimensionality of the data while preserving as much variance as possible. The cumulative energy plot shown in Figure 1 was used to determine the number of PCA components required to retain 70%, 80%, 90%, and 95% of variance.

% variance	PC-modes
70	2
80	3
90	5
95	7

Table 1. PC-modes necessary for % variance

After analyzing the cumulative energy, k=2 and k=3 were selected for visualization via plots and classification experiments. Variance drops sharply as the meaningful structure of the data is captured in the first few principal components due to being highly correlated. The effects of retaining higher PC-modes is discussed later on and is shown in Table 2

A centroid-based classification approach is implemented, where the mean position of each movement type in the reduced PCA space is found. Each sample is classified based on its proximity to these centroids using Euclidean distance. To evaluate the performance of the classifier, the training accuracy is computed by comparing predicted labels with ground truth labels. This is

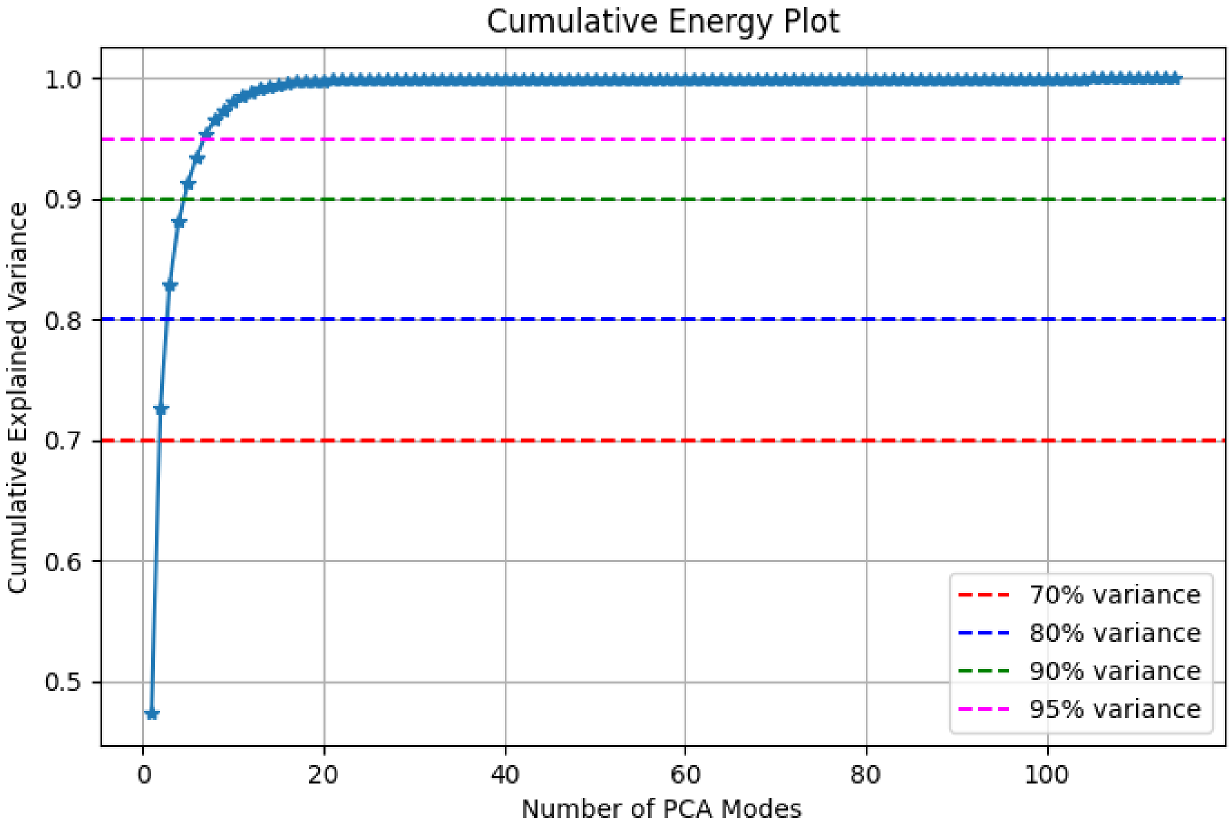

FIGURE 1. Cumulative energy used to determine number of PCA components required to keep different amount of the variance

done using sklearn.metrics [2]. The optimal number of PC-modes is determined by analyzing the classification accuracy across various values of k. Alternatively, k-Nearest Neighbors classifier was implemented and compared to the centroid-based classifier.

4. Computational Results

To analyze the separability of movement types, the first two (k = 2) and three (k = 3) principal components are visualized. Scatter plots in 2-D (Figure 2) and 3-D (Figure 3) PCA spaces show the clear clustering of different movement types. The results indicate that the three movement types exhibit distinct clustering patterns, indicating that PCA effectively captures significant movement differences.

The classifier accuracy in test samples is higher than the training accuracy in both k = 2 and k = 3 PC modes (Table 2). In 2-D PCA space, test accuracy is 10% higher than the training accuracy, and in 3-D PCA space, the test accuracy is even greater, 17%, than the training accuracy. One possible explanation is that PCA reduces noise, making the classifier more robust. However, a small test set may also contribute to this result, necessitating further analysis. This seems to be the case even when choosing higher k-values, leading to the conclusion that the model generalizes well and performs better in the test set.

To explore other classification approaches, k-Nearest Neighbors (KNN) classifier was implemented in the PCA space, using k = 3, with 3 nearest-neighbors. Comparing the performance of KNN with the centroid-based approach (Table 3) shows favor towards the KNN-method, as

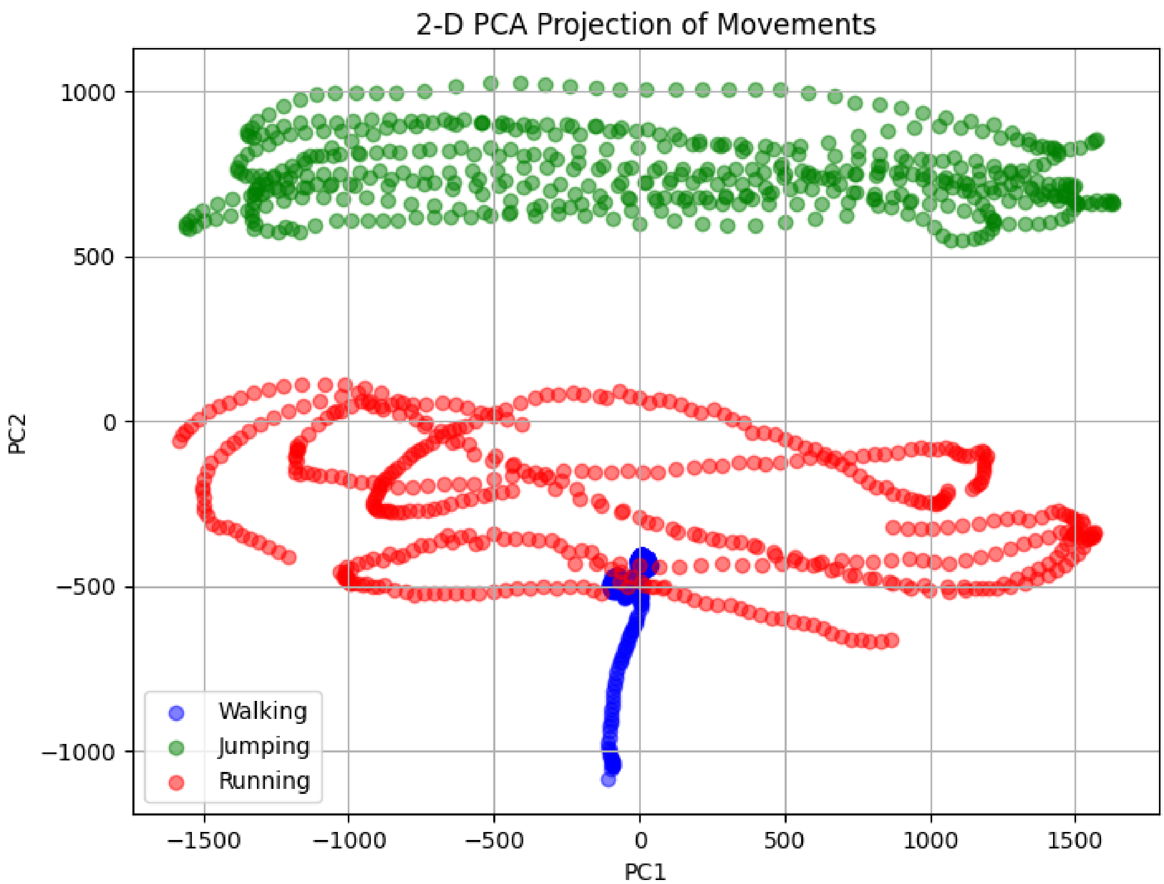

FIGURE 2. Scatter-plot in 2-D of movement types

PC-modes	% training-accuracy	% test-accuracy
2	88.13	98.33
3	75.60	92.33
5	75.07	91.67
7	87.07	94.33
8	87.53	93.00
10	88.80	94.33
16	91.07	95.33
20	91.07	95.33

TABLE 2. Higher PC-modes and the effect on accuracy

it performed much higher than the centroid-based method did in 3-D PCA space, achieving test accuracy of 100%. It should be noted that the test sample is a limited dataset, and may not be wholly representative of 'real-world' data. However, KNN is also more computationally complex and will be slower for larger datasets.

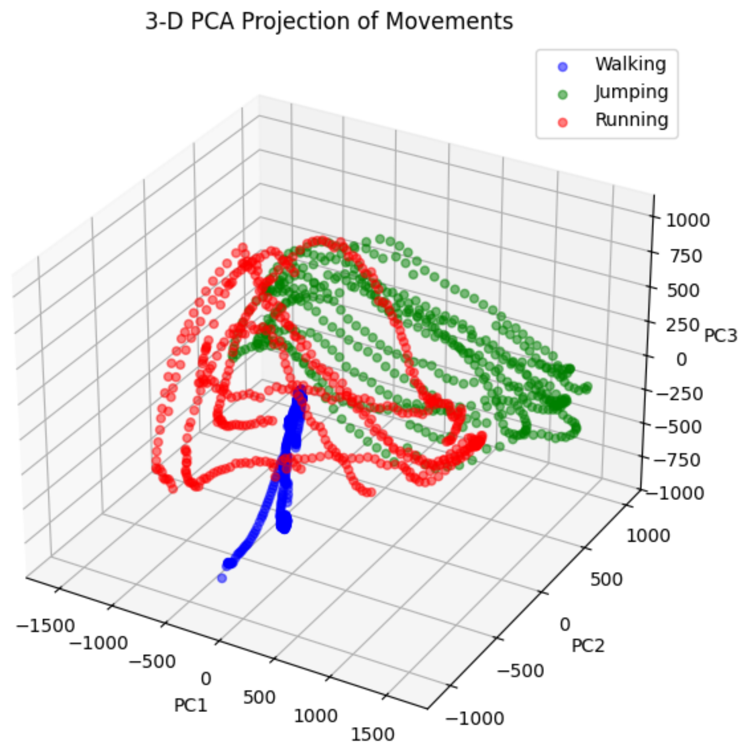

Figure 3. Scatter-plot in 3-D of movement types

classifier	% training-accuracy	% test-accuracy
centroid-based	75.60	92.33
kNN	99.87	100.00

Table 3. Training and testing accuracy of centroid-based versus K-Nearest Neighbors (K = 3) classifiers in 3-D (PCA1, PCA2, PCA3) space

5. Summary and Conclusions

The 2-D and 3-D plots suggest that lower-dimensional PCA representation is sufficient for movement classification. In humanoid movement data, there are dominant movement patterns that allow quick classification, e.g. when the leg moves during a walking motion, the knee and ankle move predictably compared to when running. Furthermore, higher values of k somewhat improve classification accuracy - capturing finer variations and any noise - but only up to a threshold, beyond which additional components provide diminishing contribution, as seen in Table 2. To assess the effectiveness of the centroid-based classification model, test accuracy is compared with the training accuracy. The classifier maintains high accuracy on test samples, confirming that PCA successfully preserves specific movement characteristics.

The results indicate that KNN performs exceptionally better than centroid-based classification, even when choosing higher k-values. KNN offers potential advantages in handling non-linearly separable data, while centroid-based classification remains computationally efficient.

This study demonstrates the application of principal component analysis for humanoid robot movement classification. Specifically, it was determined that PCA effectively reduces the dimensionality of the data while preserving critical movement characteristics. The visualization of movement in the PCA space reveals distinct clustering of different types of movement. Centroid-based classification achieves high accuracy, which confirms the effectiveness of PCA for feature extraction. The test accuracy remains consistent with the training accuracy, indicating a strong generalization. Further, KNN provides a viable alternative classification method, with performance higher than that of centroid-based classification.

Future work could include training with a larger dataset and verifying independence between test and training data. Also, there is opportunity to explore other dimensionality reduction techniques such as SVM and investigate the use of deep learning models for classification of humanoid movement.

Acknowledgements

Special thanks to Eli Shlizerman for detailed explanations regarding Singular Value Decomposition and Principal Component Analysis. The author acknowledges the assistance of computational tools for plotting and debugging.

References

[1] J. Kutz. *Methods for Integrating Dynamics of Complex Systems and Big Data*. Oxford, 2013.
[2] scikit learn. User guide - 2.5. decomposing signals in components (matrix factorization problems).