ROBOT MOTIONS, DIMENSION REDUCTION, AND CLASSIFICATION

SAMIP KARKI

Applied Mathematics, University of Washington, Seattle, WA
karki1@uw.edu

Abstract. The singular value decomposition is used in many data applications. In this work, I use singular value decomposition and principle node analysis for dimension reduction. Taking position data from a robot performing different tasks such as walking, running, and jumping, I find that the position measurements, which have dimension of 114, can be accurately approximated with as little as 10 principal component modes. I make a classifier which assigned new measurements labels (walking, running, or jumping) by comparing the distance in the principal component mode basis between the new measurement and the centroids of each of the labels.

1. Introduction and Overview

In this project, I have measurements representing a robot performing three movements: walking, jumping, and running (figure 1). The raw data comes in the form of (x, y, z) coordinates of various joint at various time-steps. I reduce the dimension of this data using using principal component mode analysis and create a simple classifier which can assign the the labels of walking, jumping, running to each time-step.

2. Theoretical Background

When analyzing data with high dimension, it is often useful easier to consider lower dimensional representation of the data. Consider a sequence of measurements represented by the columns of the $n \times m$ matrix A. The singular value decomposition of A is given by:

$$A = U\Sigma V^T.$$

Date: August 30, 2024.

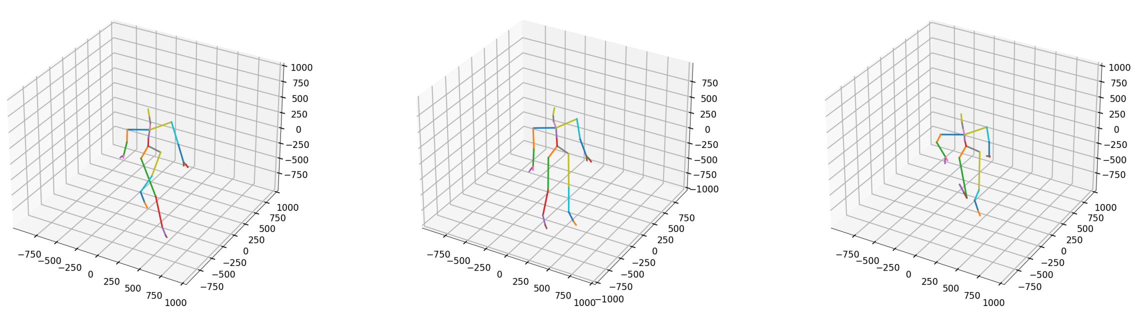

Figure 1. Examples of the each of the three labels, from left to right: walking, jumping, and running.

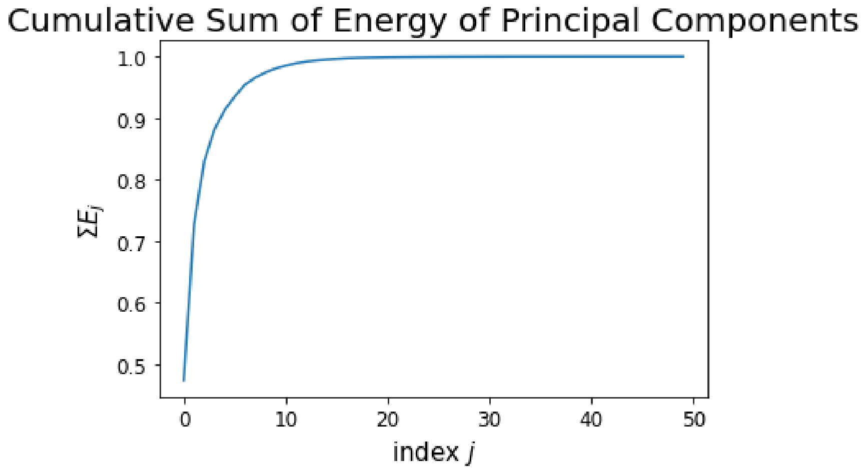

Figure 2

If $C = \Sigma V^T$, then j-th column of A is given by:

$$\vec{a_j} = U\vec{c_j} = \sum_{i=1}^{n} \vec{u_i} c_{ij}$$

where $\vec{c_j}$ is the j-th column of C, and $\vec{u_i}$ is the i-th column of U. It can be seen that the j-th measurement $\vec{a_j}$ can be expressed as a linear combination of $\{\vec{u_1}, \vec{u_2}, ..., \vec{u_n}\}$,the principal component (PC) modes, and the coefficients of the linear combination are given by $\vec{c_j}$. By truncating this sum, we obtain a lower dimensional approximation of $\vec{a_j}$.

The columns of U are ordered in such a way that the modes which contribute the most to the measurements $\{\vec{a_1}, \vec{a_2}, ..., \vec{a_m}\}$ appear first in the sum. The matrix C contains the coefficients of the measurements in the PC basis. Instead of doing analysis on A, we can instead analyze the first k rows of C, a lower dimensional approximation to A projected into the PC mode basis.

3. Algorithm Implementation and Development

All the computation and plots made in this project is done with the cannonical python packages: `numpy, matplotlib`. SVD was computed using `numpy.linalg.svd`.

4. Computational Results

Our training data consists of 500 time steps of the humanoid robot performing each movement: walking, jumping, and running. Let $\widetilde{X}_{train}$ be the 114×1500 matrix where the rows represent the (x, y, z) coordinates of the 38 joints of the robot the at each time-step. X_{train} is the matrix where each row of $\widetilde{X}_{train}$ is subtracted by the mean of that row. I take the singular value decomposition of X_{train}

$$X_{train} = U\Sigma V^T$$

to obtain the basis of PC mode $\{\vec{u_1}, \vec{u_2}, ..., \vec{u_n}\}$ and the coefficients ΣV^T.

The energy, E_i of the i-th PC mode is given by

$$E_i = \frac{\sigma_i^2}{\sum_k^{114} \sigma_k^2}.$$

The cumulative sum of the Energy of the first j PC modes of X_{train} are graphed in figure 2. The first 1, 2, 4, and 6 PC nodes are required to accumulate 70, 80, 90, and 95 percent of the total

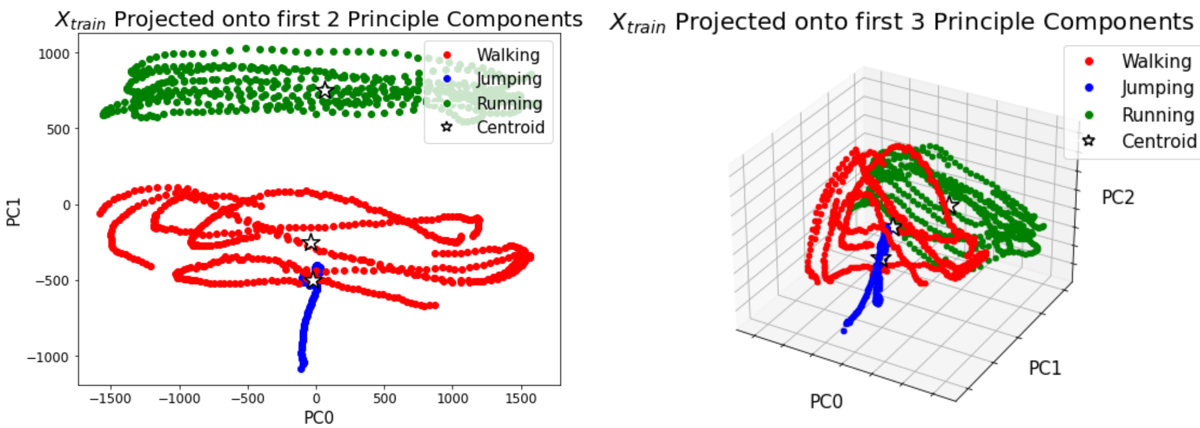

FIGURE 3. Projection of the columns of X_{train} on to the first 2 (left) and first 3 (right) principal component modes of X_{train}. The colors represent each of the three label, walking, jumping, running.

energy, respectively. It can be concluded that as low as 6 PC is all that is needed to sufficiently capture the characteristics of the measurements in X_{train}.

The projection of the columns of X_{train} onto the first 2 and 3 PC modes are given in figure 3. It can be seen that each label, walking, jumping, and running, has a discernible path which is taken.

Next, I leverage PC analysis to create a simple classifier. My goal is to take new measurements $\widetilde{X}_{test}$ and to correctly assign labels to them Y_{guess}. I find the centroids, $\vec{c}_w$, $\vec{c}_j$, and $\vec{c}_r$ of the measurements in X_{train} corresponding to each label, walking, jumping, and running, respectively, shown in figure 3. The algorithm for the classifier is as follows:

- Transform $\widetilde{X}_{test}$ by subtracting the row mean of $\tilde{X}_{train}$ from each row and change to PC basis by applying U_k^T, where U_k is truncated to only have the first k columns of U. The projected data matrix will be called $X_{test,PC}$
- For the i-th column in $X_{test,PC}$ called $\vec{x}_i$, find the L2 distance between $\vec{x}_i$ and each of the centroids $\vec{c}_w$, $\vec{c}_j$, and $\vec{c}_r$ using the first k PC mode coefficients of the centroids.
- Assign the label in Y_{guess} for which centroid $\vec{x}_i$ was closest to.
- Do this for various values of k.

On the left of figure 4, I plot how accurate the classifier is at labeling the training data $\widetilde{X}_{train}$ as the number of PC modes included k changes. On the right of figure 4, I use the classifier to label a new data set of measurements $\widetilde{X}_{test}$ which includes 100 measurements of each of the labels. We can see that the performance of the classifer does not get any better with more than 10 PC modes included. This is consistent with figure 2, where most of the energy of X_{train} was within the first 10 PC modes.

5. Summary and Conclusions

In this project, I investigated how principle component analysis can be to drastically reduce the dimension of data. In particular, when using a simple classifier based on the centroid of the training labels, I saw that my classifier only needed 10 PC modes to obtain the best results. I learned about SVD in a previous course, AMATH 584 Numerical Linear Algebra, but this project was the first time I used principal component analysis in a hands-on application. It was satisfying to take the theory I had previously learned and apply it to a practical application.

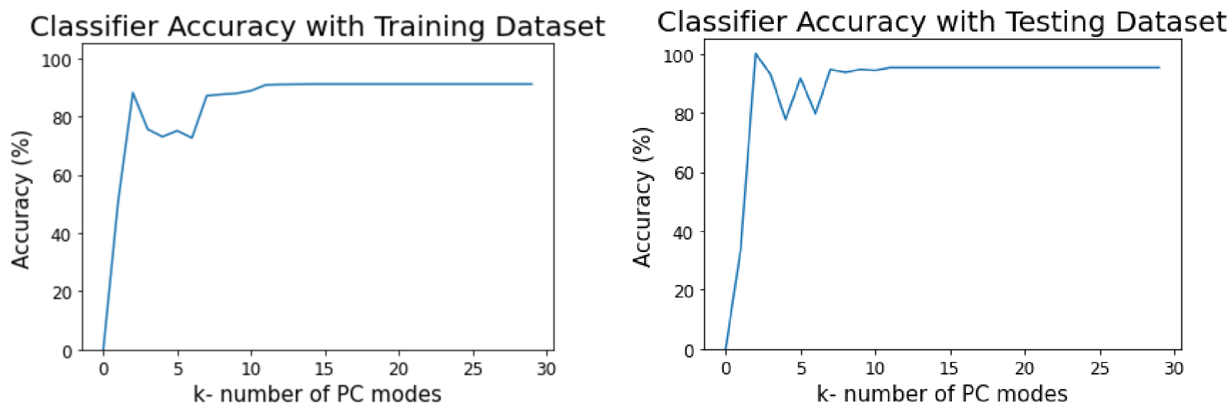

FIGURE 4. The performance of centroid distance classifier for various number of PC modes included, k. On the left, the classifier's performance in labeling X_{train}. One the right, the classifier's performance in labeling X_{test}.

ACKNOWLEDGEMENTS

I'd like to thank Prof. Shlizerman who was available to answer my questions during office hours.