**Distance Functions**

Euc\_between\_subjects: calculates Euclidean distance on the raw data for the between-subject case

Euc\_within\_subjects: calculates Euclidean distance on the raw data for the within-subject case

Maha\_between\_subjects: calculates Mahalanobis distance on the raw data for the between-subject case

Maha\_within\_subjects: calculates Mahalanobis distance on the raw data for the within-subject case

DTW\_between\_subjects: calculates DTW distance on the raw data for the between-subject case

DTW\_within\_subjects: calculates DTW distance on the raw data for the within-subject case

Var\_Euc\_between\_subjects: uses maximum variance to reduce the dimensionality of raw data, and afterward calculates Euclidean distance on the low-dimensional data for the between-subject case

Var\_Euc\_within\_subjects: uses maximum variance to reduce the dimensionality of raw data, and afterward calculates Euclidean distance on the low-dimensional data for the within-subject case

Var\_Maha\_between\_subjects: uses maximum variance to reduce the dimensionality of raw data, and afterward calculates Mahalanobis distance on the low-dimensional data for the between-subject case

Var\_Maha\_within\_subjects: uses maximum variance to reduce the dimensionality of raw data, and afterward calculates Mahalanobis distance on the low-dimensional data for the within-subject case

Var\_DTW\_between\_subjects: uses maximum variance to reduce the dimensionality of raw data, and afterward calculates DTW distance on the low-dimensional data for the between-subject case

Var\_DTW\_within\_subjects: uses maximum variance to reduce the dimensionality of raw data, and afterward calculates DTW distance on the low-dimensional data for the within-subject case

Var\_loglikelihood\_between\_subjects: uses maximum variance to reduce the dimensionality of raw data, and afterward calculates loglikelihood on the low-dimensional data for the between-subject case

Var\_loglikelihood\_within\_subjects: uses maximum variance to reduce the dimensionality of raw data, and afterward calculates loglikelihood on the low-dimensional data for the within-subject case

PCA\_Euc\_between\_subjects: uses PCA to reduce the dimensionality of raw data, and afterward calculates Euclidean distance on the low-dimensional data for the between-subject case

PCA\_Euc\_within\_subjects: uses PCA to reduce the dimensionality of raw data, and afterward calculates Euclidean distance on the low-dimensional data for the within-subject case

PCA\_Maha\_between\_subjects: uses PCA to reduce the dimensionality of raw data, and afterward calculates Mahalanobis distance on the low-dimensional data for the between-subject case

PCA\_Maha\_within\_subjects: uses PCA to reduce the dimensionality of raw data, and afterward calculates Mahalanobis distance on the low-dimensional data for the within-subject case

PCA\_DTW\_between\_subjects: uses PCA to reduce the dimensionality of raw data, and afterward calculates DTW distance on the low-dimensional data for the between-subject case

PCA\_DTW\_within\_subjects: uses PCA to reduce the dimensionality of raw data, and afterward calculates DTW distance on the low-dimensional data for the within-subject case

PCA\_loglikelihood\_between\_subjects: uses PCA to reduce the dimensionality of raw data, and afterward calculates loglikelihood on the low-dimensional data for the between-subject case

PCA\_loglikelihood\_within\_subjects: uses PCA to reduce the dimensionality of raw data, and afterward calculates loglikelihood on the low-dimensional data for the within-subject case

En\_Euc\_between\_subjects: uses autoencoder neural network to reduce the dimensionality of raw data, and afterward calculates Euclidean distance on the low-dimension data for the between-subject case

En\_Euc\_within\_subjects: uses autoencoder neural network to reduce the dimensionality of raw data, and afterward calculates Euclidean distance on the low-dimension data for the within-subject case

En\_Maha\_between\_subjects: uses autoencoder neural network to reduce the dimensionality of raw data, and afterward calculates Mahalanobis distance on the low-dimension data for the between-subject case

En\_Maha\_within\_subjects: uses autoencoder neural network to reduce the dimensionality of raw data, and afterward calculates Mahalanobis distance on the low-dimension data for the within-subject case

En\_ DTW\_between\_subjects: uses autoencoder neural network to reduce the dimensionality of raw data, and afterward calculates DTW distance on the low-dimensional data for the between-subject case

En\_ DTW\_within\_subjects: uses autoencoder neural network to reduce the dimensionality of raw data, and afterward calculates DTW distance on the low-dimensional data for the within-subject case

En\_Loglikelihood\_between\_subjects: uses autoencoder neural network to reduce the dimensionality of raw data, and afterward calculates loglikelihood on the low-dimension data for the between-subject case

En\_Loglikelihood\_within\_subjects: uses autoencoder neural network to reduce the dimensionality of raw data, and afterward calculates loglikelihood on the low-dimension data for the within-subject case

**Utility Functions**

EM\_boundingCov: learns the parameters of a Gaussian Mixture Model (GMM) using a recursive Expectation-Maximization (EM) algorithm. After each EM step, the covariance matrices are bounded to avoid numerical instability

EM\_init\_regularTiming: initializes the parameters of a Gaussian Mixture Model (GMM) by using k-means clustering algorithm

gausPDF: computes the Probability Density Function (PDF) of a multivariate Gaussian represented by means and covariance matrix

loglik: computes the loglikelihood of a GMM model

**Data Augmentation (used in the v1 version of the article)**

M1\_Augmentation: generates new instances by adding random noise to the correct instances. The input data is “M1-DeepSquat” in the folder “Data for Distance Functions”.

**Neural Networks**

The network predicts quality scores transformed from GMM loglikelihood for the between-subject case. The input to the networks are the data with 117 dimensions.

Spatio-Temporal NN is the proposed deep learning model in the corresponding article.

Spatio-TemporalNN\_M1: predicts quality scores and “Data\_Load\_GMM\_Bet” is called to load data. The fixed permutation “M1\_Shuffled\_Indices” in Data\_shuffle is used to shuffle the data.

CNN\_GMM\_Between\_M1: predict quality scores and “Data\_Load\_GMM\_Bet” is called to load data. The fixed permutation “M1\_Shuffled\_Indices” in Data\_shuffle to shuffle the data.

The same naming rule is applied for RNN.

CNN\_GMM\_Between\_M1\_Aug: predict quality scores for the augmented data and “DataA\_Load” is called to load data. The split function is used to shuffle data randomly.

The same naming rule is applied to RNN.

**Data for Distance Functions**

M1-DeepSquat-Correct: the original data performed correctly in the first exercise - Deep Squat

M1-DeepSquat-Incorrect: the original data performed incorrectly in the first exercise - Deep Squat

M1-Reduced-DeepSquat: obtained by performing dimensionality reduction with autoencoder neural networks to compress M1-DeepSquat

**Data for Neural Networks**

M1\_DeepSquat folder

Train\_X1: the raw measurements of correct movements

Train\_Y1: the corresponding quality scores for the correct movements

Test\_X1: the raw measurements of incorrect movements

Test\_Y1: the corresponding quality scores for the incorrect movements

M1\_Aug\_DeepSquat folder

X1\_movement1: the raw measurements of correct movements

Y1\_movement1: the corresponding quality scores for the correct movements

Xk\_movement1 (k=2, 3, 4, 5): synthetically generated sequences for the correct movements

Yk\_movement1 (k=2, 3, 4, 5): the corresponding quality scores for the synthetically generated sequences

X6\_movement1: the raw measurements of incorrect movements

Y6\_movement1: the corresponding quality scores for the incorrect movements